A SIMULATION APPROACH TO
GRANITE SENTRY SYSTEM ANALYSIS

THESIS

Marilyn J. Bauer, Captain, USAF

AFIT/GOR/ENS/94M-02

[redacted]

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/94M-02

Accesion For		
NTIS CRA&I		X
DTIC TAB		
Unannounced		
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

A SIMULATION APPROACH TO GRANITE SENTRY SYSTEM ANALYSIS

THESIS

Marilyn J. Bauer, Captain, USAF

AFIT/GOR/ENS/94M-02

Approved for public release; distribution unlimited

Disclaimer Statement: The views expressed in this thesis are those of the author and do not reflect the official policy of position of the Departement of Defense of the United States Government.

AFIT/GOR/ENS/94M-02

A SIMULATION APPROACH TO GRANITE SENTRY SYSTEM ANALYSIS

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operation Research

Marilyn J. Bauer, B.S.

Captain, USAF

AFIT/GOR/ENS/94M-02

March 1994

Approved for public release; distribution unlimited

THESIS APPROVAL

STUDENT: Marilyn J. Bauer CLASS: GOR-94M

THESIS TITLE: A Simulation Approach to Granite Sentry System Analysis

DEFENSE DATE: 1 March 1994

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Advisor	Lt Col Paul F. Auclair/ENS	Paul F. Auclair
Reader	Lt Col Dennis C. Dietz/ENS	Dennis C Dietz

Acknowledgements

Over the past eight months, I have received invaluable guidance, and support from many people. To begin, I wish to thank my fellow GOR/GST-94M classmates. Working with you has made this experience enjoyable. I cannot think of a better group of people with which to have completed this adventure.

Many thanks go to several people who provided essential technical assistance. To my sponsor, Capt Andy Hachman, thank you for your guidance and advice along the way. Your assistance, especially in reviewing the model and a draft of my thesis, was extremely helpful. To the folks at Martin Marietta, Eugene Bagenstos and Nate Trachta, I could not have pursued this topic without your willingness to help. I truly appreciate the fact that you always took time to find answers to my questions. To Lt Col Dietz, thank you for taking the time to provide suggestions and corrections to improve the quality of this thesis. And to Mr Jack Dunn, thank you for taking time to thoroughly review my thesis--this assistance was invaluable to me.

And finally, Lt Col Auclair, I cannot thank you enough for the advice and guidance you have given me these past eight months. Thank you for being my mentor, my advisor, and especially my friend. Your patience and dedication in assisting me in this effort will be remembered and appreciated always.

I look forward to working with all of you in the future as I continue my Granite Sentry endeavors in Colorado Springs.

Marilyn Bauer

Table of Contents

List of Figures

Figure Page

List of Tables

AFIT/GOR/ENS/94M-02

Abstract

This study demonstrated the use of simulation mòdelling to analyze Granite Sentry system performance. The availability simulation model constructed provides a number of system performance measures as a function of component MTBFs and MTTRs. Analysis of failure data prior to model construction supported the generally accepted use of exponentially distributed failure rates and lognormally distributed repair times. A Microsoft Windows version of SLAMSYSTEM proved to be an efficient modelling tool, especially during early stages of model development. Guidelines for model use in system analysis are explored through a runtime analysis and a response surface model of system downtime as a function of part redundancy. The runtime analysis provides recommendations for appropriate simulation runtime and number of replications to produce reasonably efficient and accurate results. The response surface analysis highlights three system components whose part redundancy significantly affects system downtime. Finally, the analytical availability model developed was an essential validation tool in simulation model development.

A SIMULATION APPROACH TO
GRANITE SENTRY SYSTEM ANALYSIS

I. *Introduction*

1.1 Background

Simulation modelling has been recognized as essential in evaluating system performance, especially when cost or asset constraints limit the testing of actual systems (19:553). One-of-a-kind systems pose especially stringent asset constraints in that there is generally not a system available for dedicated, extensive testing. The asset constraints encountered when testing one-of-a-kind systems suggest distinct advantages of developing simulation models of these systems. Simulation models used during testing could be categorized as availability simulation models since one of their primary purposes would be to estimate of system performance measures related to, and including system availability. Model estimation of numerous system and component level reliability, maintainability, and availability (RM&A) statistics could aid decision makers in determining system limitations, and design configurations.

1.1.1 Literature Review. The articles reviewed below illustrate the use of simulation modelling in system analysis. Specifics of model construction were system dependent and therefore are not explained in detail. All articles reached similar conclusions related to use of simulation modelling as a tool in conducting system

analysis--all researchers were avid supporters of simulation modelling and believed their results were enlightening because of the simulation. In addition to the examples below, the Air Force Test and Evaluation Center (AFOTEC) has used simulation to aid in determining system availability for several systems, including F-16s (9).

Brown developed a simulation model of Mobile Satellite Command Systems to analyze endurance and availability (5:3). His model simulates a generic system which is considered representative of all similar systems. The model is sufficiently flexible to incorporate varied system configurations and fully exploit sensitivity analysis (5:1-3). His use of experimental design and systematic sensitivity analysis is most relevant to this research effort.

Pohl used a simulation model to evaluate F-15E availability during operational testing (19:549). She modeled a squadron of F-15Es (24 aircraft) under both peacetime and wartime scenarios. Due to the inherent impossibility of testing under wartime conditions, and the exorbitant cost of testing an entire squadron, she concluded that "Use of the F-15 Availability Model was essential to the evaluation of F-15 mission reliability, maintainability and availability during OT&E"(19:553). Using this model, she estimated the squadron RM&A statistics based on test data from only two aircraft (19:553). The ability to evaluate a system under a wide range of scenarios not possible through actual testing is one of the major advantages of using simulation in RM&A studies.

Several researchers note advantages of simulation in their research. In his article on use of simulation in availability calculations, Schroeder cites the work of

other analysts (Naylor, Moore and Clayton, and Mogenthaler) to develop the following list of benefits of simulation as an analysis tool.

Simulation:

1) allows thorough study of complex interactions within a system.

2) is the means to analyze effects of varying external factors such as environment and other connected systems on system behavior.

3) allows for determination of critical variables and levels through sensitivity analysis.

4) can be used to experiment with new situations that may occur and about which there is little or no collected information.

5) serves as a "no risk" test prior to implementation of the system for testing new policies and/or decision rules.

6) enables study of dynamic systems in compressed time.

7) allows the breakdown of a complex system into subsystems providing the means for additional understanding and analysis of the system.

8) can help predict bottlenecks or weak points in the system. (21:746-747)

Both Pohl's, and Boyd and Bavuso's analyses add support to Schroeder's advocacy of simulation. In Pohl's discussion of her model, she clearly states that simulation was "essential" to evaluating F-15E RM&A measures of performance. In her report, she lists some of the same simulation benefits as Schroeder (19:550). Boyd and Bavuso similarly agreed that simulation is beneficial and, in fact, essential to complete analysis. They also emphasized a need for co-existing analytical tools that are compatible. They stated these co-existing tools are necessary to give the analyst sufficient flexibility in conducting the analysis (4:106, 112). This thesis supports this

finding through development of an availability spreadsheet model to be used for system analysis in addition to the simulation model.

Simulation allows for a thorough analysis of system performance under a variety of scenarios and system conditions. A simulation-based analysis can provide great insight into system performance, especially when extensive testing of actual systems is not possible, making simulation a useful analysis tool.

1.1.2 Model Requirement. The Studies and Analysis division of the Headquarters of the Air Force Test and Evaluation Center (HQ AFOTEC/SAL) plans to use simulation more frequently in system analysis, especially during testing of one-of-a-kind systems (9). Further, HQ AFOTEC/SAL has been tasked to develop a generalized availability simulation model for analyzing one-of-a-kind systems. Since construction of this generalized model will likely require about two years, AFOTEC personnel suggested that developing an availability simulation model for a specific system could assist them in completing their task (9). Additionally, construction and use of an availability simulation model for a specific system could aid in evaluation of that system.

1.2 Objectives

The purpose of this research was to develop an availability simulation model of the Message Processing Section (MPS) of Granite Sentry, a computer system in the Cheyenne Mountain Complex. The model calculates intrinsic system availability, including only computer hardware and software component contributions to

availability. Additional model outputs include mean time between critical failures, component availabilities, as well as many other RM&A statistics. This research also displays the capability and benefit of developing and using an availability simulation model in system analysis.

1.3 Background of CMU and Granite Sentry

The false attack indications issued by the defense system in Cheyenne Mountain in Nov 1979 and June 1980 caused great panic throughout the nation (13:3). These events clearly demonstrated the need to upgrade obsolete computer systems in the Cheyenne Mountain Complex (CMC). The Cheyenne Mountain Upgrade (CMU) Program began in 1981 to modernize several aspects of the CMC such as the Communications Systems Segment (CSS), the NORAD Computer System (NCS), the Mission Essential Back-Up (MEBU), and the Command Center Processing and Display System (CCPDS) (13:3). Granite Sentry was added to the CMU Program in July 1985 to improve a variety of attack warning and assessment missions by enhancing computer systems that relay air and missile warning messages from sensors outside the CMC to the workcenters within (13:3).

The Granite Sentry upgrade is a phased acquisition of both hardware and software replacements. System enhancements to date include installation of seven new Digital Equipment Corporation (DEC) VAX 8550 computers, functioning as the main system processors, and thirty new DEC VAX 3540 workstations. The final phase will include additional hardware and software upgrades to further improve

processing of air and missile warning information (13:4). Final operational testing of the Granite Sentry system is currently scheduled to occur in 1995 and will complete the phased installation of the system.

1.4 Granite Sentry System Description

The following sections describe the specific hardware and software components of the Granite Sentry system and their functional relationships. Understanding these relationships is essential to constructing a useful model.

1.4.1 Hardware. Figure 1.1 shows the three major hardware components in Granite Sentry (GS): workstation displays, a gateway and processor cluster, and a communications network. The system has been designed with component redundancies which are used to increase hardware availability. This redundancy is important since component availability is the primary factor used in calculating system operational status (13:5).

Currently, there are 30 DEC VAX 3540 workstations located in workcenters throughout CMAFB, although the GS software can support up to 64. Workstations display a processed version of national defense air and missile messages transmitted from sensors outside CMAFB (13:5).

The gateway and processor cluster is a group of computers linked together to share resources (disk drives, storage devices, databases). The GS cluster is composed of seven DEC VAX 8550 processors. Two function as gateways for data to enter and exit Granite Sentry, while five serve as mission processors that analyze data brought in

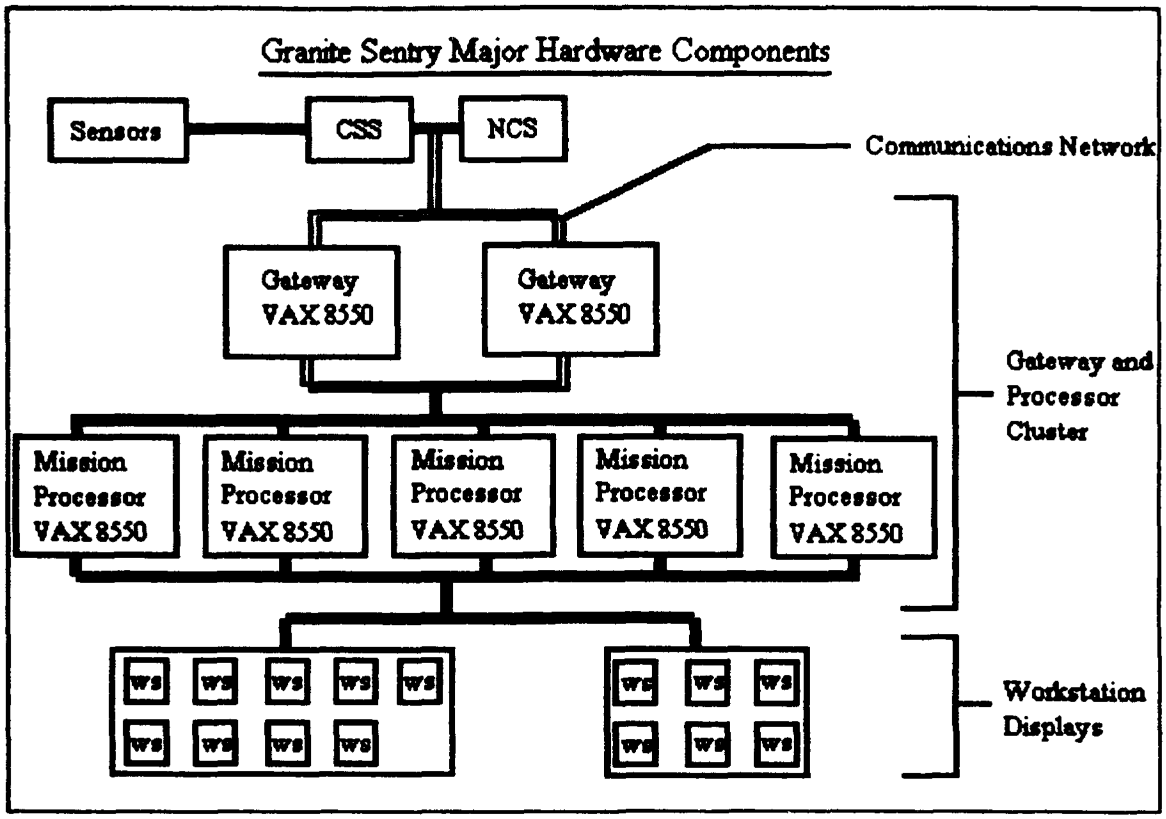


Figure 1.1: Granite Sentry Major Hardware Components

through the gateway servers. After processing is complete, the data is ready for transmittal to the workstations via the ethernet (13:5).

A dual rail ethernet provides a communication link between all hardware components. The ethernet consists of two separate data transmission lines running in parallel. The two separate paths available for data flow provide backup transmission capability throughout the system (13:6). Only one functional line is necessary to transmit data through the system.

1.4.2 Software. Granite Sentry software is organized in functional groups called Computer Software Configuration Items (CSCIs). The six current CSCIs in

Granite Sentry are Gateway, Command Post, Air Mission, Workstation, System, and Support (13:6). Table 1.1 gives a brief description of CSCI functions, how they are interconnected, and with which hardware components they are associated.

1.4.3 System Input and Output. Input to Granite Sentry consists primarily of messages containing national defense information about air or missile attacks. Air messages come into Granite Sentry from the Communications Systems Segment (CSS) and relay threat information that is of an "air breathing" nature. Missile messages come to Granite Sentry from the NORAD Computer System (NCS) and transmit information related to possible missile threats.

Output from Granite Sentry is displayed on workstation video displays. Correct processing of the "raw" sensor messages and transmittal of this information to CMC workcenters via their workstations is the mission of the Granite Sentry system.

CSCI NAME	CSCI FUNCTION	HARDWARE ASSOCIATION
Gateway CSCI	provides the interface between GS and the sensors; receives data from the CSS and NCS, writes it to a journal, and passes it on to the appropriate mission CSCI	VAX 8550
Command Post CSCI	handles message processing and validation for all missile warning data from the Gateway CSCI; processes missile warning data manually input from workstation keyboards	VAX 8550
Air Mission CSCI	handles all message processing and validation for all air defense data from the Gateway CSCI; transmits messages back to the Gateway CSCI to be passed to the outside world; processes air defense data input from workstation keyboards	VAX 8550
Workstation CSCI	displays processing for missile warning and air defense data; receives data from both the Command Post and Air Mission CSCIs	VAX 3540 workstations
System CSCI	provides "behind-the-scenes" software capabilities such as system monitoring, maintenance functions, system control, clock control, and message journaling; most functions used by all other CSCIs	VAX 8550 and VAX 3540 workstations
Support CSCI	provides data analysis and exercise/simulation capabilities; allows user requests for reports of all the incoming and outgoing data from GS	VAX 8550 and VAX 3540 workstations

Table 1.1: Granite Sentry Software

II. Definitions and Model Assumptions

2.1 Definitions

2.1.1 Reliability. The probability a system performs its required mission (i.e. works for a specified period of time) is the system mission reliability (24:1). Mission reliability is used to describe systems functioning with "hot" backups that immediately take over when the primary system fails (24:2). Since Granite Sentry systems meet the definition requirements of mission reliability, reliability calculated in this thesis are mission reliabilities.

Reliability statistics can be calculated by obtaining failure information about system components, calculating the amount of time between failures (TBFs), and fitting this data to a distribution. Parameters of this lifetime distribution describe the failure rate of the system which can be translated to system reliability. Complexity of these calculations occurs when system components are both in parallel and in series and when failure of a component does not necessarily translate to system failure (4:6). Granite Sentry is composed of a combination of series and parallel structures.

Failure rates can also be calculated using the following relationship (12:6):

Failure Rate = # failures/Total Operating Time

2.1.2 Maintainability. Maintainability of a system is defined "...as the probability that, when maintenance is performed under specified conditions, the system will be up again (in a state of operation) within a specified period" (24:10).

Maintainability is determined through calculation of the Time To Repair (TTR) a component after each failure. A collection of TTRs for a specific component can be graphed to comprise a TTR probability density function for the component. The parameters of this frequently lognormal distribution describe the repair rate, and thus determine the maintainability of the component (4:6).

2.1.3 Availability. Combining reliability and maintainability measures yields system availability which is usually of primary interest to the system user. Availability is the probability the system is able and committable to performing its required mission(s) at any random time. It can be calculated by a ratio of system "up time" divided by total system operating time ("up time" + "down time") (12:7).

Availability can also be calculated directly from the reliability and maintainability distribution parameters (20:440). It is well known that as operating time approaches infinity, estimations of system or component availability can be made knowing only the mean of the failure rate and repair rate distributions--the type of distribution need not be known (20:441). This fact provides the basis for availability calculations made in the analytical model presented in section 3.3 of this thesis (20:441).

Availability of a Series System with i components:

$$A=\prod_{i}\frac{\mu_i}{\mu_i+\lambda_i} \tag{1}$$

where μ_i = repair rate of component i, and λ_i = failure rate of component i

Availability of a Parallel System with i components:

$$A=1-\prod_{i}\frac{\lambda_i}{\mu_i+\lambda_i} \tag{2}$$

For the purpose of thesis calculations, the term ***model availability*** is used to refer to numbers resulting from calculations using formulas 1-3 above. This ensures that availability statistics generated are interpreted as being a direct result of the modelling assumptions under which they were created.

2.2 Model Assumptions

2.2.1 System Structure Assumptions. The following are assumptions related to the specific structure used in the simulation model.

1) The system is modeled as the Initial Operational Capability (IOC) model, rather than the Final Operational Capability (FOC) model, since information about the IOC model was most readily available. The FOC model was still in its planning stages.

2) Redundant parts within components are simulated as being accessed in numerical order. (eg. Messages are sent first to gateway hardware processor number 1. Only if this unit is down are messages sent to the 2nd processor.) This is a method of simplifying the simulation model structure and was verified as an acceptable model assumption by Martin Marietta personnel (2).

3) All redundant parts are assumed to serve as "hot standbys"--ready for immediate use. No crossover time is modeled when a back-up part is used.

4) Subsystems are independent--failures in one component do not affect the other components.

5) The system is assumed to be in continuous operation.

6) Maintenance resources are assumed to be available in unlimited supply and immediately upon part failure.

2.2.2 Critical and Non-Critical Failure Assumptions. The following assumptions define system failure classifications.

1) Components for which a failure does not constitute a critical failure are not included in the model. Non-critical components for Granite Sentry include:

Miscellaneous Hardware (including printers, ethernets)
Miscellaneous Software
Personnel who cause system failure through procedural error

2) The *critical path* refers to the minimum hardware and software components required to ensure adequate message transmission. A *critical failure* is any failure that violates one of these minimum requirements. This critical path, obtained from Martin Marietta, is composed of the components listed in Table 2.1 (2).

2.2.3 Other Assumptions

1) The system is not taken down for training--training is completed off-site.

2) Data saves/back-ups are done while the system is operational.

3) New software (except for new operating systems which are loaded about every two years) is loaded while the system is up.

4) All system hardware and software components are assumed to be running all the time--no down time other than for repairs.

5) Distributions for failure rates will be obtained from actual data and are assumed to already incorporate a "burn-in" period therefore represent "steady state" performance. This "burn-in" period is not accounted for separately by the simulation model. Due to the memoryless quality of exponential failure rates (which are assumed in the model) and the high availability of components, it is assumed that starting simulation runs with no failed components is acceptable.

6) Scenarios for message flow (air and missile messages) are based on general statements made by Martin Marietta personnel. Actual scenarios used by Martin Marietta for testing are classified.

COMPONENT NAME	CRITICAL NUMBER
Power Distribution Unit (PDU)*	1 of 2
Gateway (DEC VAX 8550)	1 of 2
Air Gateway Software	1 of 2
Mission Gateway Software	1 of 2
Star Couplers*	1 of 2
HSC 70 Disk Controllers	1 of 2
RA82 disk drives (arranged in 10 shadow sets containing 2 drives each--1 of 2 must function)	8 of 10 shadow sets
Mission Processors (DEC VAX 8550)	1 of 5
Air Mission Software	1 of 2
Command Post Software	1 of 2
Status Monitor Software	1 of 5
Workstation hardware	3 of 6 (ADOC) 7 of 9 (NCC)
Workstations software	3 of 6 (ADOC) 7 of 9 (NCC)
* indicates no failures have occurred to date	

Table 2.1: Granite Sentry Critical Components

III. Model Development

This chapter begins with discussion of the background steps completed prior to simulation model construction. Following this background section is a detailed explanation of the Granite Sentry simulation model in words and diagrams. The chapter concludes with description of an analytical spreadsheet model developed as an additional analysis tool.

The following terminology is used throughout discussion of the Granite Sentry models.

component - refers to one of the 13 critical components of Granite Sentry listed in Table 2.1. (eg. power distribution unit, gateway, etc.)

part - smallest independently functioning portion of a component (eg. Gateway 1, SC 1, etc.)

message processing - messages transmitted to Granite Sentry from outside sensors are subsequently "processed" by hardware and software components that interpret and configure data in a useful format for Granite Sentry operators

messages received - messages successfully processed by a component are considered to be received by that component; messages considered to be received by the system have been successfully processed by all system components

messages not received - messages not processed by a component due to critical failure of that component are considered as not received by both the part and the system

The following terms are used to organize simulation model discussions:

segment - portion of the simulation model that performs a specific independent function (eg. Hardware Failure Segment, Message Creation Segment, etc.)

section - refers to segmentation of the simulation model based on independent network flow--entities travel only within a section (eg. Air and Missile Message Processing Section, etc.)

3.1 Background to Model Construction

This section reviews steps taken prior to formulation of the Granite Sentry availability simulation model. Extensive analysis and understanding of the system was essential to producing an accurate representation of the system in the simulation model.

3.1.1 Determination of Model Requirements. Model requirements were determined through communication with the contractor, Martin Marietta, and with AFOTEC personnel (1, 10, 22). Discussions with Martin Marietta personnel indicated that calculation of system availability was of primary importance. AFOTEC also suggested that the following elements, or at least their possible effects on availability, should be included in an availability simulation model (10:4):

1) Hardware element
2) Software element
3) Human element
4) The physical environment--interaction with other equipment
5) Effect of sparing levels and maintenance concepts

Due both to lack of time and necessary information, the system as modeled in this thesis only directly reflects the first two elements.

Recommendations for model output included:

1) Mean Time Between Critical Failures (MTBCF)
2) Mean Time to Repair (MTTR)
3) Intrinsic Model Availability
4) Mean Time Between Maintenance (MTBM) (scheduled and unscheduled)
5) Mean and 90th percentile hardware repair time

3.1.2 Determination of System Structure and Model Components. Model components were determined through review of Martin Marietta white papers (11, 12) and the Granite Sentry Technical Support Manual (13). Component structure was also obtained from these sources and verified through telephone communication with Martin Marietta personnel (2).

Granite Sentry is composed of the 13 critical hardware and software components listed in Table 2.1. These components function together to process air and missile messages received from sensors outside CMAFB. In addition, Granite Sentry software generates a type of "status message" revealing the status of the system to its operators. Though transmission of these status messages is not critical to system functioning, lack of transmission indicates a failure of the status monitor software which, in turn, is considered a critical system failure. Accurate depiction of the hardware and software components that process air and missile messages is the basis of the Granite Sentry availability simulation model.

3.1.3 Analysis of Component Failures. Copies of the Joint Reliability and Maintainability Evaluation Team (JRMET) Meeting Minutes, provided by Martin Marietta, contained failure data for Granite Sentry from September 1991 to August 1993. To facilitate analysis, data from 1992 was transcribed to a Microsoft EXCEL spreadsheet format. In the interest of time, the 1992 sample was considered sufficiently representative of most components; however, all data would have been used in the analysis if data could have been obtained in electronic format. Appendix A contains the spreadsheet file of failures sorted by component type.

3.1.4 Fitting Failure Data to Lifetime Distributions. An EXCEL spreadsheet file containing all failure data sorted by component facilitated calculation of times between failures and times to repair. For components whose failures were identified by specific part (eg. workstation 1 hw), TBFs were calculated for each part and then aggregated to develop a sample database. Parameters of distributions fit to this sample database characterize each part and were used to generate part failures in the simulation model. TTRs were aggregated in a similar manner to develop a database used to fit distributions representing part TTRs.

When failure data did not include part references, TBFs were calculated for the entire component (eg. Gateway hw). Parameters of distributions fit to this data characterize the *component* and were transformed to yield parameters of the *part* failure distribution (eg. Gateway 1) through assumption that component failures can be evenly distributed between the parts. In other words, if a component with 2 parts had 8 failures, then each part was assumed to have 4 failures. For exponentially distributed failure rates, this transformation consisted of dividing the component failure rate by the number of parts, yielding a part failure rate. The parameters of the part failure rate distributions were used in the simulation model to accurately represent components as their individual parts and not simply as one big "black box".

Once the TBF and TTR databases for each component were complete, Palisade's BestFit software was used to actually fit the data to a distribution. The specific procedure entailed:

1) Copying raw TBFs for a component into a BestFit sample data file.

2) Displaying the data using the Input Distribution plot (histogram format).
3) Setting the number of classes in the histogram to the maximum that allowed a "smooth" histogram with no empty intervals--see Figures 3.1 and 3.2.
4) Invoking BestFit's distribution fitting algorithm--see Figure 3.3.
5) Collecting statistics on raw data and Lack of Fit tests for the fitted distributions.
6) Selecting a "best" distribution type to represent the TBFs or TTRs based on comparison of Kolmogorov-Smirnov, Anderson-Darling, and Chi-Square statistics.

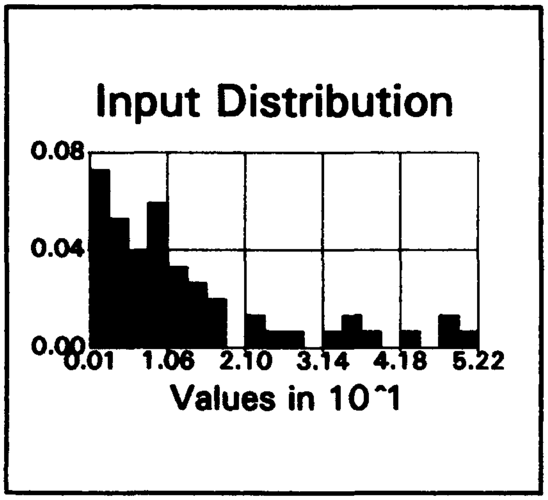


Figure 3.1: Unacceptable Histogram--Empty Intervals

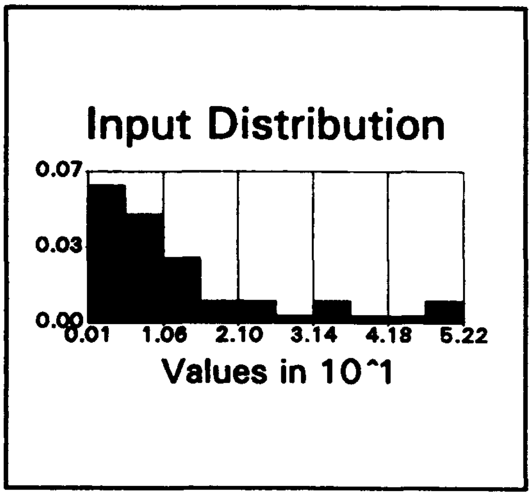


Figure 3.2: Acceptable Histogram--NO Empty Intervals

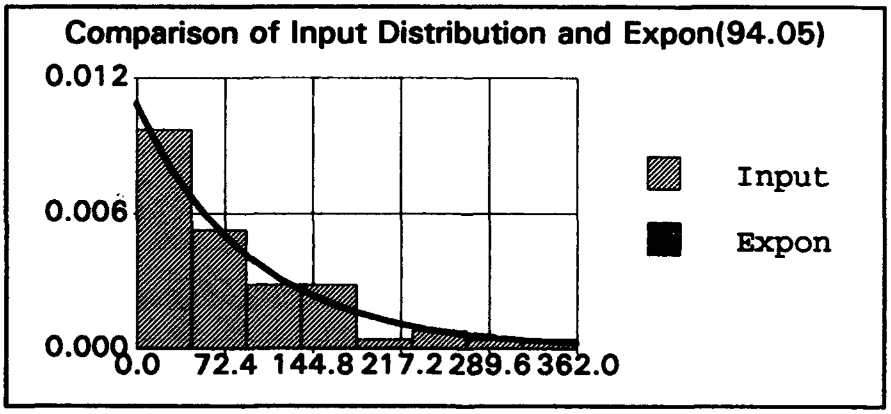


Figure 3.3: BestFit Comparison of Sample Data to a Fitted Distribution

3.1.4.1 Lack of Fit Tests. Bestfit presents rankings of the fitted distributions based on three goodness-of-fit tests: Chi-Square, Kolmogorov-Smirnov, and Anderson-Darling. These tests are briefly described below.

The Chi-Square Test "measures how well the sample data fit a hypothesized probability density function" (17:23). This is a test based on "...a comparison of observed cell counts with their expected values under the hypothesis..."(6:63) By making this type of cell count comparison, this test ignores some information contained in the data making it "less powerful" than other tests in some situations. The Chi-Square test has wide applicability because of its generalized approach, but sensitivity to choice of cell size renders it inappropriate for small samples (14:82).

The Kolmogorov-Smirnov Test *(K-S Test)* compares "...an empirical distribution function with the distribution of the hypothesized function"(17:23). It measures the vertical distance between a step function created by the sample data and the hypothesized distribution. The test statistic is calculated based on the largest observed vertical distances between the two distributions (6:100).

The Anderson-Darling Test *(A-D Test)* is "...designed to detect discrepancies in the tails of the distributions"(17:23). Similarly to the K-S statistic, it measures the differences between the empirical and hypothesized cumulative distributions, but the differences are squared in calculation of the A-D test statistic. Calculation of the A-D test statistic also includes a weighting function that assigns larger weights to differences obtained from comparison of the tails of the distributions (6:100).

For this analysis, results of the Chi-Square test are sufficient to select a representative distribution where sample sizes are large (workstation hardware and software failures). For the other components, whose failure data was limited, the K-S and A-D tests are more appropriate since these test statistics do not depend on assignment of sample values to histogram cell counts (14:81-83). Additionally, since this research involves coarse estimation of failure and repair distributions, the K-S test will be preferred to the A-D test. Since the calculation of K-S test statistics does not involve adding extra weight to the tails of the distribution, the K-S test appears more appropriate for coarse distribution estimation.

3.1.4.2 Bestfit Results. A summary of Bestfit's selected distributions, their parameters, and goodness-of-fit test statistics are listed by component in Appendix B. Test statistics revealed that the general assumption of exponentially distributed failure rates is not unreasonable. When the exponential distribution did not have the best fit, the parameters of the better fitting gamma or weibull distribution rendered their distribution approximately exponential. Similarly, values of K-S test statistics for distributions chosen as best fits to the TTR data revealed that the assumption of lognormally distributed failure times was not unreasonable.

3.2 Availability Simulation Model Construction

Model requirements, system structure, and component characteristic distributions were the basis for developing the availability simulation model. Subsection headings below outline the general process used to develop the final model.

3.2.1 Initial Simulation Model Construction. A Microsoft Windows version of SLAMSYSTEM was used to construct an initial simulation model of the system. In many ways, this method of network coding was very efficient. The pictorial, menu-driven method of model construction allowed a focus on correctly modelling the system structures rather than on coding details. Similarities between the representation of different model components allowed for extensive use of block copying. Additionally, having a pictorial representation of the physical system to refer to in model validation was extremely useful. Tracking entity flow along a picture is generally easier than tracking entity flow through program code.

Because of similarities in component structures, model construction was completed in segments, with many segments repeated extensively throughout the model. Explanation of the network will follow a similar segmented approach. To begin with , the network is divided into three main sections. These sections are referred to as the Air and Missile Message Processing Section, the Status Message Processing Section, and the Time-Based Random Failure Generation Section. Block diagrams at the beginning of each section depict the general processes conducted within the section. Within sections, segmentation occurs and is described in detail. SLAMSYSTEM diagrams of each

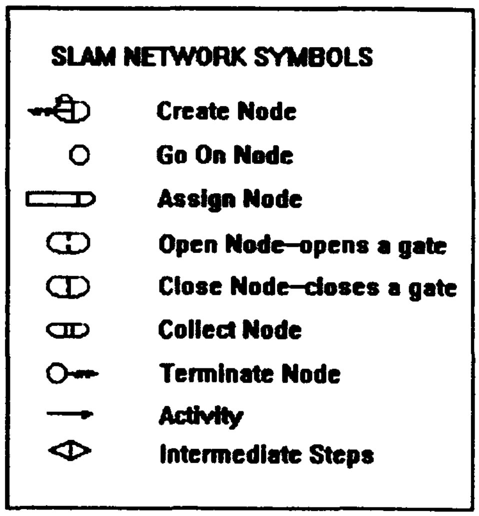


Figure 3.4: SLAM Network Symbols

segment are included to facilitate understanding of segment processes. Figure 3.4 defines the standard SLAM symbols used to diagram the segments.

Air and Missile Message Processing Section

This section contains all the hardware and software components required to process either air or missile messages. These components include all system critical components except for the status monitoring software. This section of the network was created using the two segments outlined below. Figure 3.5 displays a general block diagram outlining the basic processes represented in this section.

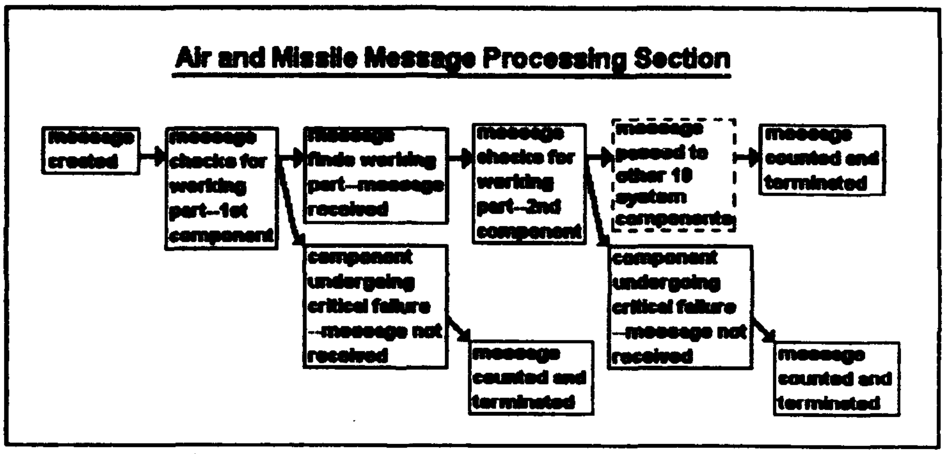


Figure 3.5: Air and Missile Message Processing Section

Segment 1: Message Creation

The air and missile messages transmitted to Granite Sentry from sensors outside CMAFB are simulated as SLAM entities. The length of time between creation of message entities is based on unclassified generalizations of message flow rates obtained from Martin Marietta (1). Actual testing scenarios are classified and were not obtained or used.

Upon creation, message entities are assigned a 1 or 2 to attribute 2 defining it as either an air or missile message. The upper create node in Figure 3.6 corresponds to air messages; the lower to missile messages. After the message entity passes through an activity, counters are incremented to add up the total number of air and missile messages created. From these assign nodes, message entities are passed to a go-on node which simulates the "door" to Granite Sentry. Once the message entities pass through this node they are processed by the appropriate Granite Sentry system components.

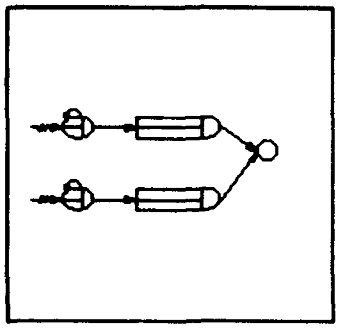


Figure 3.6: Air and Missile Message Creation

Segment 2: Component Representation

This segment is used to represent all system components in this section. As seen in Figure 3.7, the segment starts with a go-on node with the number of emanating activities equal to the number of component parts plus one. These activities are conditioned on the status of gates that represent part availabilities. When the gate is open the part is working and message entities can flow through that activity. Message entities arriving at the first go-on node will choose the first open activity, checking them in sequential order (see NOTE below). After passing through the activity, the message entity arrives at a collect node which tallies the number of messages processed by that part.

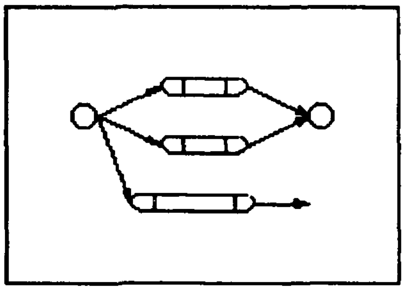


Figure 3.7: Component Representation

If the component is experiencing a critical failure, no parts are functioning and all gates are closed. If all gates are closed, message entities are sent to a collect node tallying all messages not processed by this system component. It is assumed that messages must be transmitted immediately upon receipt with no delay acceptable. In reality, the message would have been logged in the computer journal and could be received; however, doing so requires replay of the tapes covering message flow during the time period the system was down, which is not done in practice (2). All messages successfully processed by this component are passed to the next component for processing. Twelve of these segments exist in the model.

NOTE: In all cases, message entities start "checking" gates in numerical order, checking the gate for the second part only if the first part gate is closed. This branching protocol is a simulation modelling convenience and does not necessarily refer to actual labels of physical components.

Time-Based Random Failure Generation Section

This section generates all failures observed in the system. (The only exceptions are message-based failures occurring in the Status Monitor Software Segment; however, these failures never actually occur in model runs because no input data for distribution of these failures was available.) The block diagram representing basic failure simulation is shown in Figure 3.8.

Segment 1: General Time-Based Part Failures

This segment is the heart of the gate system and is used to activate/deactivate parts of the system based on simulated failures. This segment generates failures for

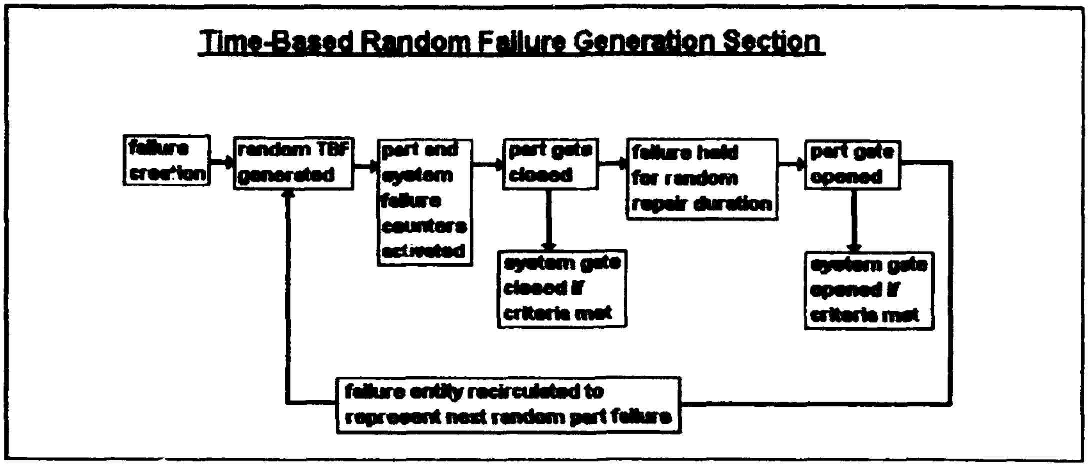


Figure 3.8: Time-Based Random Failure Generation Section

one part of a component (eg. gateway 1). Failure rates and MTTRs, calculated as indicated in section 3.1.3, are the basis for failure entity creation and failure durations in this segment. Figure 3.9 diagrams the segment.

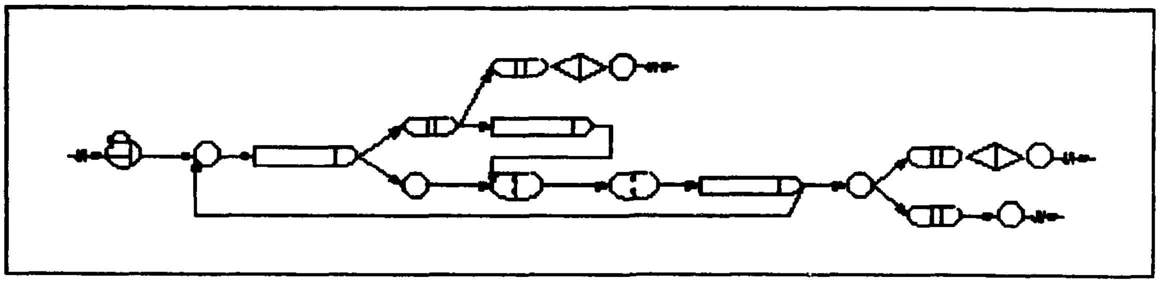


Figure 3.9: Time-Based Part Failures

At time zero one failure entity is created. It first travels along an activity to a go-on node where the failure simulation process begins. From the go-on node, the failure entity travels along an activity for an exponentially distributed amount of time with a mean equal to the parameter of the failure rate distribution for that part. Once the failure passes through this activity, it causes the first part failure. The failure

entity then passes through an assign node activating counters for part failures and total system hardware/software failures. Additionally, at this assign node, a failure duration is randomly generated and assigned to the failure entity as an attribute. After passing through the assign node, the failure entity sent along one of two paths depending on whether a component critical failure has occurred.

If a component critical failure *has not* occurred, the failure entity passes through a go-on node and closes the part gate. After closing the gate, it passes along an activity of duration equal to the time to repair, and then re-opens the part gate. The failure entity then passes through an assign node which decrements the counter for current component part failures. Upon leaving the assign node, the failure entity is cloned into entities A and B. Entity A either travels to a collect node and then to open the system critical failure gate, if appropriate, or is just collected. In either case Entity A ends up being terminated. Entity B is recirculated back to the first go-on node to represent the next part failure. Recirculation of failure entities was necessary to avoid creation of first failures for all parts at time zero.

If a component critical failure *has* occurred, the failure entity passes through a collect node counting critical component failures. Next, the failure entity is cloned into entity 1 and 2. Entity 1 passes through the system critical failure collect node, closes the system critical failure gate, and is subsequently terminated. Entity 2 passes through an assign node where attribute 4 is assigned a value marking the failure entity as a critical failure. From here the failure entity is sent to close the part gate and continues to be processed as outlined above when a component critical failure has not occurred.

Segment 2: RA 82 Failures

As observed in Figure 3.10, RA 82 disk drives are structured differently than any other major component. There are 20 disk drives paired into 10 shadow sets. In other words, 2 disk drives operate in parallel, and as long as one of these two drives is operational, the shadow set is considered functional. Additionally, 8 of the 10 shadow sets must be functional at any one time. Less than 8 functional shadow sets represents a system critical failure. Failures in this segment represents failure of an entire shadow set.

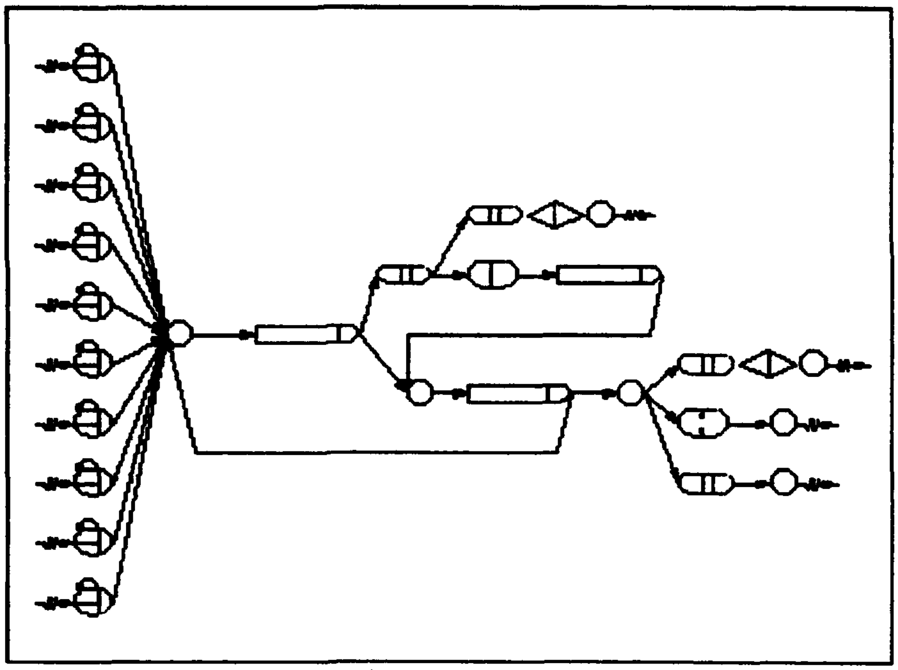


Figure 3.10: RA82 Disk Drive Failures

Ten failure entities are created a time zero and represent complete failure of one of the 10 shadow sets. As in the previous segment, failures first pass through a go-on node and then through an activity of exponential duration with a mean equal to the parameter of the failure rate distribution for one shadow set. The failure entity then activates counters for total system hardware failures, total RA82 failures, and current RA82 failures. After incrementing the counters, the failure entity

can take one of two paths depending on whether a component critical failure has occurred.

If a component critical failure *has* occurred, meaning that more than 2 shadow sets have failed, the entity passes through a collect node that tallies the number of RA82 critical failures. Next, the failure entity is cloned forming entities 1 and 2. Entity 1 passes through the system critical failure collect node, closes the system critical gate, and is subsequently terminated. Entity 2 closes the component gate and is assigned a value to attribute 4 marking it as a critical failure. From here, entity 2 rejoins the non-critical failure processing loop at the second go-on node and is processed as described in the next paragraph.

If the component critical failure *has not* occurred, the failure entity passes along an activity to the second go-on node. The failure entity leaves the go-on node along an activity with duration equal to the length of the failure stored in its attributes. Once the failure is over, the failure entity passes through an assign node causing the current failure counter to be decremented. From here, the entity is cloned into entities 1 and 2. Entity 1 returns to the first go-on node to begin the next shadow set failure. Entity 2 is passed to the next go-on node. From the go-on node, entity 2 takes at most two of three paths. If the failure was a system critical failure and the component is now functional, the entity takes two paths. The two paths enable it to simultaneously travel through a collect node to open the system critical failure gate and to open the component part gate. If the failure was not critical, the entity is collected as a non-critical failure and then terminated.

Segment 3: Workstation Failures

In Figure 3.11, the top create node represents creation of a workstation hardware failure entity, the bottom create node represents creation of a software failure entity. The combination of these create nodes therefore represents one workstation. Additional workstations are simulated by adding another create node feeding into the each of the first go-on nodes.

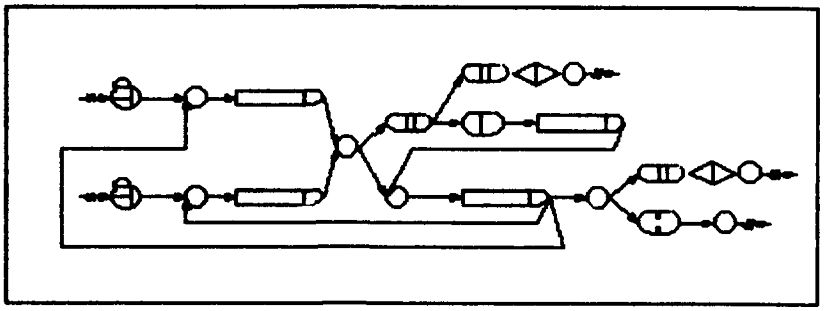


Figure 3.11: Workstation Failures

As in previous failure segments, one entity is created by each create node. It passes through a go-on node, and then travels along an activity of exponential duration. After passing through this activity, attribute 2 is assigned a 1 or 2 defining the failure as either a hardware or software failure. Additionally, at this assign node counters for workstation hardware and software failures as well as for current component failures are incremented. From here, both types of failure entities travel to the same go-on node and are subsequently processed along the same node sequence.

Flow from the second go-on node in this segment follows exactly as described in the RA 82 failure section, starting when RA 82 failure entities leave the first assign node. The only difference is the number of concurrent failures causing a component critical failure.

Status Message Processing Section

This section simulates the status monitor software and its failures. The message-based failure segment found in this section is included as a demonstration of the capability to include this type of failure; however, since no data was available to simulate message-based failures, message-based failure gates remain open throughout the simulation. If desired, this segment could be added to any of the software component representations. Figure 3.12 gives an overview of the processes involved in this section.

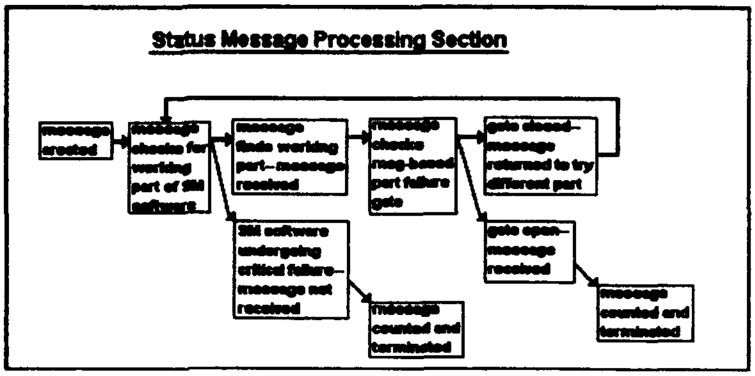


Figure 3.12: Status Message Processing Section

Segment 1: Status Message Creation

Message entities are created here exactly as in the air and missile message creation segment. The time between creation was set to a constant (10 minutes) as recommended by Martin Marietta personnel (1). The assign node contains a counter that totals the number of status messages sent.

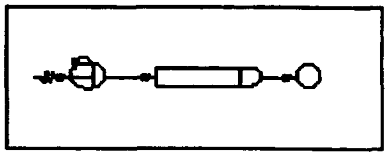


Figure 3.13: Status Message Creation

Segment 2: Message-Based Failures

This section includes the addition of a second set of failures caused by message receipt which are the basis of this segment. It seems likely that a majority of software

failures would occur and be recognized at the time of message receipt. In other words, unless the software is actually accessed, failures are not likely to be recognized. To account for such a failure mode, this segment incorporates a second software failure distribution based on message flow rather than time. However, with no data available to approximate such a distribution, these message-based failures never actually cause part failures to occur.

The only difference between message-based and time-based failures is in failure initialization. Since the failure is message-based, it occurs based on the approaching message's time of creation. If the time of message entity creation is at or beyond a randomly generated time for first message-based part failure, the message entity moves through an assign node and then closes the part gate (if it is not already closed due to a time-based failure). The entity then passes through an activity with duration equal to the lognormally distributed repair time, opens the part gate (unless a time-based failure is still occurring), and is terminated. If the time of message entity creation is before the time set for the first message-based part failure, the message entity is considered processed and is counted and terminated.

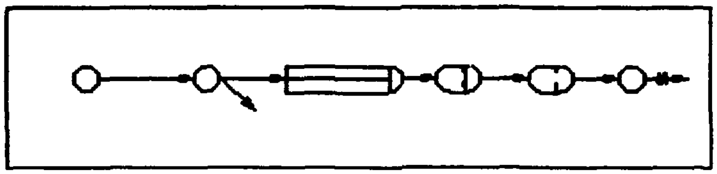


Figure 3.14: Message-Based Failures

Segment 3: Status Monitor Software

This segment is, out of necessity, a combination of the above message-based failure segment and the component representation segment from the air and missile

message section. Each part of this component experiences both message-based and time-based failures. Message entities passing through this segment check the status of

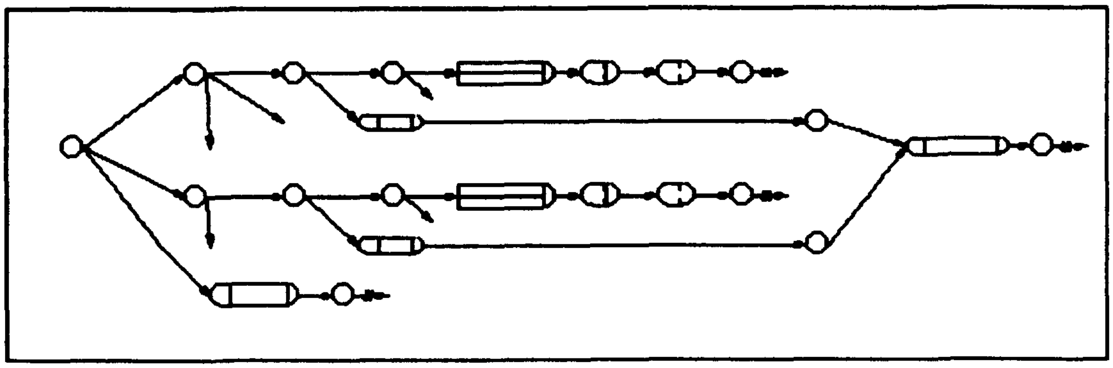


Figure 3.15: Status Monitor Software

the part gate. If the part gate is open, the entity's time of creation is compared with the time randomly selected for the first message-based part failure. If the entity's time of creation is before the time selected for the first message-based failure, the message entity is processed by that part. On the other hand, if the part gate is open but the entity's time of creation is at or beyond the time selected for the first message-based failure, the entity is cloned. The original entity checks the remaining part gates and attempts to be processed by another part of the component. If the message entity does not find a part with an open gate it is collected as not processed and terminated. Simultaneously, the clone of the entity represents the first message-based failure and as such follows the message-based failure segment flow as outlined above.

3.2.2 Model Translation to SLAM II Code. Limitations of the DOS version of SLAMSYSTEM such as a maximum of 25 gates and 50 collect nodes required that the SLAMSYSTEM model be split into two networks that could run independently. To

circumvent these limitations, the model was converted to SLAM II code that could be run on a mainframe computer. The SLAM model code was extracted from the SLAMSYSTEM network and control files and merged into one file using WORDPERFECT as a file editor. The resulting file contained useable SLAM II code which ran without error after electronic transfer to mainframe computers at the Air Force Institute of Technology. However, after cursory review of the code, SLAMSYTEM's disregard for a clearly organized structure was noted and the code was reorganized into the logical sequence presented in Appendix D. Reorganization allowed for collocation of related sections and addition of explanatory comments. This detailed review of the code revealed a few hidden errors that were corrected and s[illegible]ted inclusion of additional methods of information collection.

3.2.3 Model Verification/Validation. Determination of valid system representation was accomplished through telephone conversation with Martin Marietta personnel both after the initial model was built and after the SLAM II version was created (2). Additional model validation was completed by comparing system performance measure estimates from simulation runs with estimates calculated with the analytical availability model described in section 3.3.

Initial model construction was completed using equal failure rates and failure durations for all component parts. This essentially made all components with the same number of parts function identically. Comparing the numbers of failures per part, the numbers of messages processed, and other information collected allowed for basic verification of the model structure and entity flow.

Once these test models seemed to function appropriately, failure rates and repair times representative of actual system components were input. Review of results from these simulation runs allowed for detection of any indications of model misrepresentation. The extensive use of global variables and collect nodes found in the model structure are a direct result of model verification procedures.

3.2.4 Final SLAM II Model Development. After initially adapting the SLAMSYSTEM model to SLAM II code, a reexamination of the output requirements resulted in addition of several global variables used as counters for hardware, software, and system critical failures. Additionally, a gate was added that opens and closes based on system critical failures to provide for direct calculation of model availability. With these additions, the simulation model was considered ready for use in system analysis.

3.3 Analytical Model Construction

An analytical availability model of Granite Sentry was built as a tool for simulation model validation. Model construction was based on standard availability formulas for parallel and series systems with repair (20:440-444). The basic equations used to calculate component and model availabilities are described in section 2.1.3. The model was implemented in spreadsheet format to facilitate variation of input parameters.

Construction of the model followed directly from the simulation model structure. The same model components were included and the system was structured

with all components functioning in series. Model availability calculations were based entirely on the means of the failure rate and repair time distributions obtained from failure data analysis (20:441). The model spreadsheet is included as Appendix D.

Calculation of model availability began with estimation of availabilities for each part. Part availabilities were combined into an overall component availability. Component availabilities were then multiplied together, since components are modeled in series, resulting in overall model availability. Construction of the model, once availability formulas were understood, took only a few hours, a small fraction of the time needed to build the availability simulation model. This type of quick model availability estimation technique could be particularly useful in early stages of system testing.

IV. Guidelines for Model Use

Usually, simulation models are constructed to help answer specific questions about system performance. In this case, AFOTEC simply requested an availability simulation model for Granite Sentry without posing any specific system performance questions. This chapter, therefore, presents guidelines for model use based on possible areas of interest. First, a simulation runtime analysis provides a basis for determining the runtime and number of replications needed to produce simulation results that are largely free of initialization bias and have sufficiently small estimates of standard error. Second, a response surface method (RSM) approach could be used to explore any system performance measure as a function of model input parameters. To illustrate the method, RSM techniques are used to explore system downtime as a function of part redundancies.

4.1 Runtime Analysis

When a simulation model is used to perform system analysis, selection of runtime and number of replications can significantly impact analysis results. Since component failures and repair times are generated by random variables, the output is also a random variable and as such yields only an estimate of system performance (24:97). The inherent randomness of simulation output clearly suggests the need to consider performing multiple replications.

When questioned about appropriate simulation runtime, analysts at AFOTEC/SAL suggested a runtime estimate of 20-25 times longer than the longest MTBF contained in the model (9). Though they had no concrete basis for this estimate, their suggestion could be interpreted to correlate with Nelson's runtime recommendation of "...much longer than the initial-transient period (say 20 times longer to be concrete)" (16:13). The initial-transient period Nelson refers to is the amount of time it takes for the simulation to reach a steady-state. Though this simulation model assumes steady-state failure and repair rates, the completely operational state of the system at simulation start indicates a possible initialization bias. For an availability simulation model, Nelson's initial-transient period could be defined as the longest MTBF in the system which would lead to the same runtime estimate recommended by AFOTEC. The following analysis attempted to provide additional insight into an appropriate tradeoff between the level of simulation effort, defined by the runtime and number of runs, and the precision of the estimated system performance measures for an availability simulation model.

4.1.1 Procedures. The selection of runtimes used for comparison in the analysis began with AFOTEC/SAL's recommendation of 20 times longer than the longest MTBF(3). The longest MTBF for any Granite Sentry component is approximately 250,000 minutes. Thus, the initial estimate of appropriate runtime was 20 times 250,000, or 5 million minutes. Since a runtime less than the longest MTBF allows little chance for *any* failure of that component to occur, the shortest runtime for the analysis, 2 million minutes, was arbitrarily chosen to be between 250,000 and 5

million minutes. Three other runtimes of 10, 20 , and 30 million minutes were chosen to expand the range of times for comparison. This range simulates system uptimes between 4 and 57 years which was considered wide enough to observe any runtime effect on estimation of system performance measures.

The variability of system performance measure estimates is largely controlled through the number of replications. In his article on *Designing Efficient Simulation Experiments*, Nelson suggests the number of replications should be "...at least 2, and 10 or more if possible" (23:128). In keeping with Nelson's recommendation and reasonable amounts of computer time, 10 runs were completed for each selected analysis runtime. Data collected from each of the test runs included total system critical downtime and total number of system critical failures--two measures of system performance.

Prior to data analysis, system downtimes and numbers of critical failures were converted to rates per million minutes of runtime (1.9 years) as means to compare results across runtime groups. For example, if the system experienced 5 critical failures during a runtime of 5 million minutes, the result was translated to 1 failure in 1 million minutes. System downtimes were similarly translated. Translated results are listed in Appendices F.1 and F.2.

The runtime analysis that follows is based on the calculation of means and variances for each runtime group. For each group, a mean and variance was calculated for averages of two to ten observations. For the two run average, results of the first two experimental runs were averaged. For the three run average results of the

first three runs were used, and so on up to the 10-run average. Variances were also calculated for the same groups of data as estimates of th population variance for averages of the same number of runs. Based on these basic mean and variance calculations, the following three plots were constructed to analyze the data for each runtime group. The acronyms listed in parentheses are used to refer to the plots in the analysis discussions.

1) *One Sigma Confidence Intervals on the Means* (CIM) plots depict one sigma confidence intervals around the means for batch sizes between 2 and 10 runs for each runtime group. The sample variance (s^2) for each batch size estimates the population variance (σ^2). Since s^2 is an estimate of the variance for each random observation, the standard error of the mean (se) is calculated as

$$s_{\bar{x}} = \sqrt{Var(\bar{X})} = \frac{s}{\sqrt{n}} \tag{1}$$

where Xbar = average of n observations of a system performance measure
s = standard deviation of the population
n = number of replications averaged

2) *95 Percent Confidence Interval* (CI) plots display the effect of increasing replications on the standard error of the mean for 10 to 30 replications. The sample means and variances for 10-run batch size were used in lieu of actual means and variances for greater than 10 replications due to time constraints. (For the 30MMR group alone, it required 5 days on the VAX to process the 10-run sample.) The confidence intervals were calculated under the assumption that these 10-run means and variances are adequate estimates of the population parameters for downtime per

million minutes. Ninety-five percent confidence interval limits are calculated as in equation 2 with n equal to the batch size.

$$mean_{10} \pm \frac{s_{10}}{\sqrt{n}} * 1.96 \qquad (2)$$

where $mean_{10}$ = 10-run mean for each runtime group
s_{10} = 10-run standard deviation

3) *Standard Error of the Difference* (SED) plots reflect 95 percent confidence intervals for the difference between two sample means obtained from runs made with different system configurations. These plots again use the 10-run estimate of standard error to calculate the standard error of the difference. Ninety-five percent confidence limits are calculated using the estimates of the standard error of the difference (sed) as calculated in equation 3. Note that, for means based on n observations, the standard error of the difference in two means (equation 3) is proportional to the standard error of the mean (equation 1).

$$s_{\overline{x}_1-\overline{x}_2} = \sqrt{var(\overline{X}_1-\overline{X}_2)} = \sqrt{2*var(\overline{X})} = \sqrt{2}*\frac{s_{10}}{\sqrt{n}} \qquad (3)$$

where $Xbar_1$ = average of n observations of a system performance measure for system 1
$Xbar_2$ = average of n observations of a system performance measure for system 2

The following acronyms are used to refer to runs from each runtime group: 2 million minutes--2MMR, 5 million minutes--5MMR, 10 million minutes--10MMR, 20 million minutes--20MMR, and 30 million minutes-- 30MMR.

4.1.2 Downtime Results. Means and variances of observed downtimes are used to produce the one sigma plots of the standard error around the means (CIM) in Figures 4.1 through 4.5. The 2MMR and 5MMR plots (Figures 4.1 and 4.2) reflect large fluctuations in both means and variances for averages of less than 7 runs. These

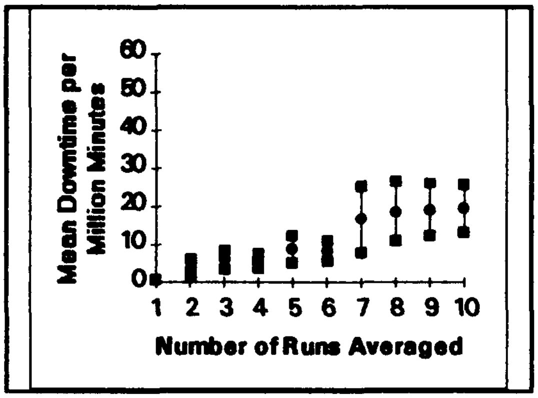


Figure 4.1: Downtime CIM--2 Million Minute Runtime Group

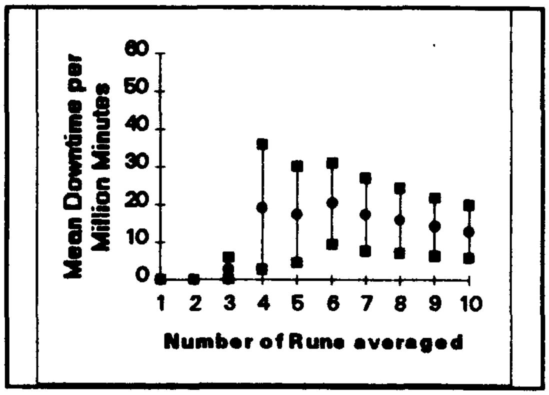


Figure 4.2: Downtime CIM--5 Million Minute Runtime Group

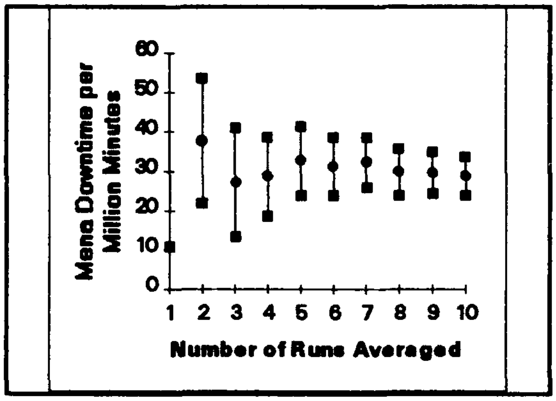


Figure 4.3: Downtime CIM--10 Million Minute Runtime Group

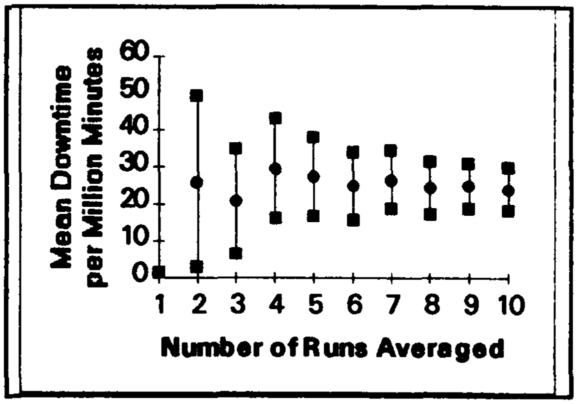


Figure 4.4: Downtime CIM--20 Million Minute Runtime Group

fluctuations are attributable to runtime in that the likelihood of observing either no failures or very few failures is greater for relatively short runtimes (ie. it is possible to obtain bad point estimates). Comparison of the 10MMR and 20MMR group plots

reveals similarities both in means and variances, indicating little added gain from doubling the runtime. The 30MMR plot shows the effect of extending runtime to this extreme (57 years of simulated time) is a significant reduction in variation even for small numbers of runs.

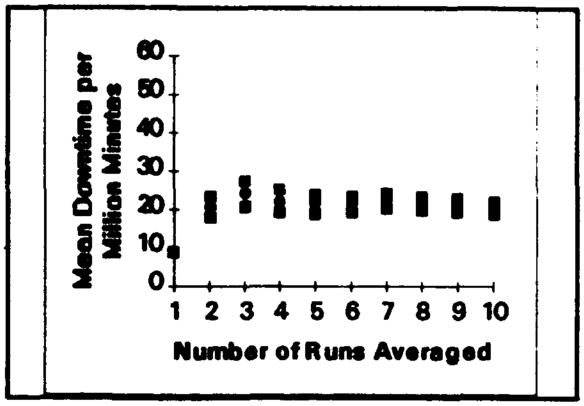


Figure 4.5: Downtime CIM--30 Million Minute Runtime Group

Results from plots in Figures 4.1 through 4.5 should also be compared to the average downtime (per 1 million minutes) calculated with the analytical spreadsheet model--7.422 minutes. The 10-run averages for all runtime groups were higher than 7.422 and ranged from 12.07 (2MMR) to 28.76 (20MMR). This likely indicates a bias in the simulation model. To put this bias in perspective of model availability, a difference in mean downtime of 25 minutes represents a difference in model availability of .000025. Since system downtime can only be estimated with confidence to 4 or 5 decimal place accuracy anyway, this difference is negligible. Therefore, results were considered accurate enough to provide a basis for this runtime analysis.

Figures 4.6 through 4.8 display plots of the 95 percent confidence intervals (CI) for mean system downtime for averages of 10 to 30 runs calculated as explained in section 4.1.1. X-axes reflect the number of runs averaged, Y-axes reflect the mean downtime per million minutes. Plots for the 5MMR and 20MMR groups are included in Appendix F.4.

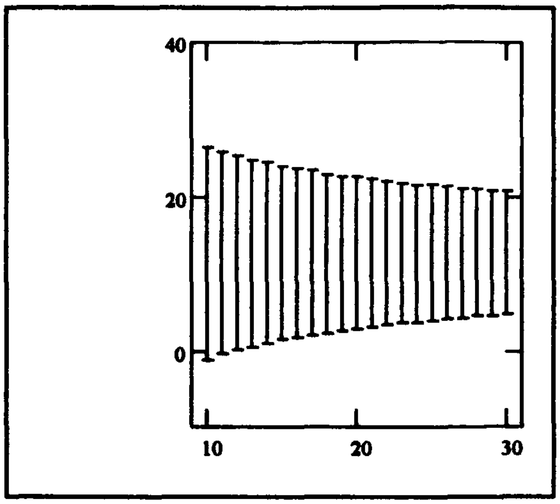


Figure 4.6: Downtime CI--2MMR

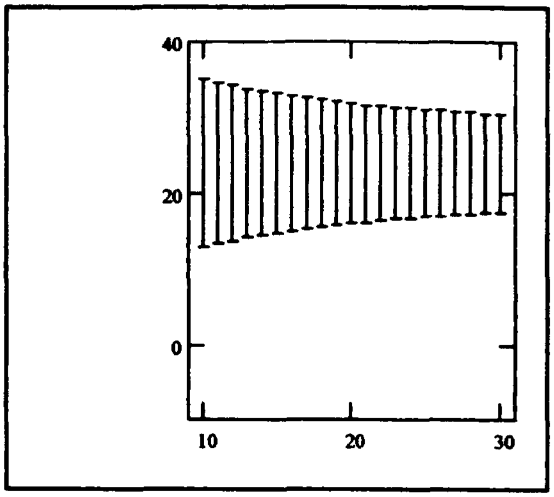


Figure 4.7: Downtime CI--10MMR

Confidence intervals displayed in Figures 4.6 through 4.8 generally support an inverse relationship between runtime and standard error. Standard error for 10-run averages decreases from approximately 7 minutes for the 2MMR group to 1.7 minutes for the 30MMR group. In terms of model availability, this represents a difference of .0000053. If model availability is accurate only to five decimal places, a difference of .0000053 is negligible.

Plots of the standard error of differences (SED) depict the standard error of the difference between two samples obtained from runs made with different system configurations. SED plots reflect decreases in the width of the 2σ confidence intervals directly proportional to those illustrated in

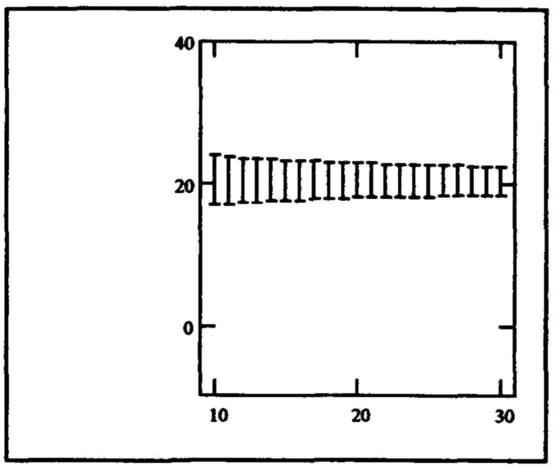


Figure 4.8: Downtime CI--30MMR

Figures 4.6 through 4.8. As an example, Figure 4.9 is the SED plot for the 5MMR group. This plot indicates that if the difference between 15-run downtime averages resulting from two different system configurations is less than 15, we cannot reject the hypothesis that these samples came from the same population. In other words, we cannot reject the hypothesis that the two system configurations produce the same performance results. This type of analysis could be particularly useful in early stages of system development.

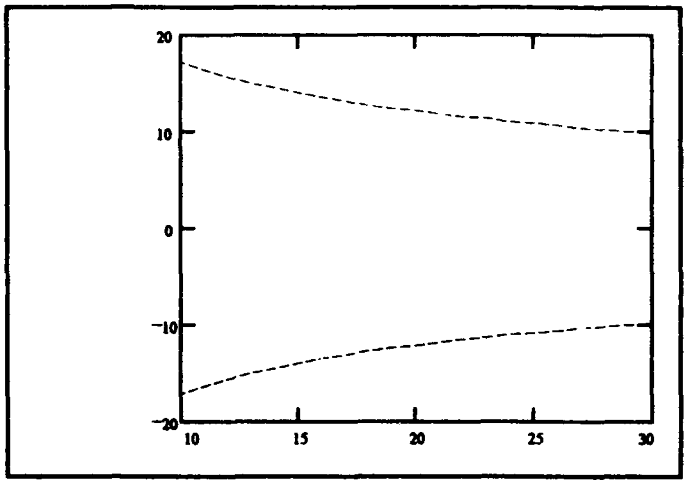


Figure 4.9: Downtime SED--5MMR

4.1.3 Critical Failure Results. Analysis of the mean number of critical failures produced plots almost completely identical (except for units) to the downtime results. Similarly to downtime results, the average number of critical failures fell consistently higher than the .0808 per million minutes average calculated with the analytical model. Averages ranged from .25 (2MMR) to .49 (10MMR). In terms of model availability, .5 failures per million minutes represents a .00005 decrease in model availability. Similarly to the simulation bias noted in downtime estimates, this bias likely does not significantly impact results. All three types of analysis plots constructed with the critical failure data are included in Appendices F.6 through F.8.

4.1.4 Summary of Results. A recommendation for the appropriate number of replications is based primarily on results displayed in the CIM plots. Consistently across runtime groups these plots suggest that an average of at least 6 or 7 runs is needed to reach fairly stable means and standard errors. Completion of additional runs beyond 6 or 7 should be based on desired accuracy of results. The smaller the allowable level of error, the greater the number of runs that must be simulated.

A recommendation for the appropriate runtime is not as clear. In general, plots indicate a reduction in variance with increases in runtime, independent of number of runs. To illustrate the selection of an appropriate runtime, suppose an estimate of system downtime (per million minutes of uptime) is required to be within 10 minutes of the actual downtime with 95 percent confidence. Referring to the CI plots, this translates to a requirement of running 19 runs for 2 million minutes, 15 runs for 5 million minutes, 13 runs for 10 million minutes, or less than 10 runs for the 20MMR and 30MMR runtime groups. Given this example, the recommendation would be to run either the 5MMR or 10MMR combinations in an effort to minimize the level of simulation effort. Recommendations resulting from this analysis coincide closely with a conclusion reached by Whitt in his article on "The Efficiency of One Long Run Versus Independent Replications in a Steady-State Simulation" (23:663). He concludes that for systems that reach a steady state quickly, "...many independent replications can be much more efficient than one long run.." (23:663). In general, it appears that sufficient accuracy can be obtained from the average of 6 or more replications with runtimes 20 times longer than the longest MTBF. Additional

accuracy can be obtained most efficiently by increasing the number of replications which in turn decreases variability.

4.2 Response Surface Analyses

Response surface analysis allows for exploration of the relationship between a system performance measure (response) and any number of controllable system input parameters (factors). These types of relationships can be characterized by a regression equation resulting from analysis of data obtained from a specific set of experimental runs. Entire courses are devoted to understanding selection of an appropriate set of experimental runs (experimental designs) and therefore will not be discussed in detail here. The interested reader can refer to Box and Draper's text, *Empirical Model Building and Response Surfaces*, for a thorough discussion of this process (20).

Generally, response surface methods (RSM) of analysis require:

1) Selection of an experimental design.
2) Selection of factors and their levels.
3) Completion of the simulation runs and collection of observed results.
4) Calculation of regression coefficients and corresponding significance levels.
5) Diagnostic testing of the regression model suggested by step 3 to determine adequacy of fit.
6) Selection of a parsimonious regression model.

Two response surface analyses were selected as appropriate in analysis of the Granite Sentry system: analysis of system critical downtime as a function of component MTBFs and as a function of part redundancies. Regression equations resulting from these analyses could be used to predict average system critical

downtime based on any combination of component MTBFs or of part redundancies included in the design regions.

4.2.1 System Downtime Analysis--Component MTBFs. Prior to selection of an experimental design, the effect of variation in component MTBFs on system downtime was coarsely studied through calculation of upper and lower bounds on system downtime. These bounds were calculated, using the analytical model, by setting all component MTBFs at their respective upper and lower 95 percent confidence interval limits. Lower and upper average annual downtime bounds were calculated as, respectively, 1.6 minutes and 10 minutes. A difference in bounds of only 8.4 minutes of system downtime per year suggested response surface development was unnecessary. The variance of the computer output alone, due to the randomness of the output, would likely mask differences of this magnitude and result in calculation of a constant model. This small difference in bounds also indicated the possibility of excess redundancy in system components.

4.2.2 System Downtime Analysis--Part Redundancies.

The above analysis, as well as basic knowledge of extensive part redundancies, suggested analysis of the relationship between part redundancies and average system downtime. Upper and lower bounds on system downtime were calculated to determine the range of results possible. The upper bound, 14,927.04 minutes annually, represents average annual system downtime for a system with no part redundancies. The lower bound, 3.68 minutes annually, represents a system with complete designed part redundancies. This wide range suggested the feasibility of a response surface analysis

since it appears that some part redundancies have a significant effect on average annual system downtime.

A two-level saturated factorial design of Resolution III was chosen for the analysis since no interactions between components were expected in this system (20:148-154). Part redundancies for all 13 critical system components were chosen as factors. Factor levels were set as representative of no redundancy (low) and complete designed redundancy (high) and are listed in Table 4.1. Specific experimental design and downtimes resulting from the 16 simulation runs are listed in Appendix G.1.

Component Name	Low Level Uncoded (Number of parts)	High Level Uncoded (Number of parts)
Gateway	1	2
Mission Gateway Software	1	2
Air Gateway Software	1	2
Mission Processor	1	5
Air Mission Software	1	2
Command Post Software	1	2
Status Monitor Software	1	5
Power Distribution Unit	1	2
HSC 70 Controller	1	2
Star Coupler	1	2
RA82 Disk Drive-shadow sets	8	10
ADOC workstations	3	6
NCC workstations	7	9

Table 4.1: Uncoded Levels for Experimental Design

STATGRAPHICS was used to perform an ANOVA and a least squares regression analysis of the system downtimes resulting from the experimental runs. The ANOVA table summarizing the effect of including all factors in the model is displayed

in Table 4.2. This initial analysis revealed three significant factors: number of Mission Gateway Software parts, number of ADOC workstations, and number of NCC workstations. The other factors had significance levels ranging from .2015 to .9229 and were deleted from the model as statistically insignificant to avoid over-specification of the model .

ANOVA for Average Annual Downtime - 13 factor study

Effect	Sum of Squares	DF	Mean Sq.	F-Ratio	P-value
A:GW	18747	1	1.8747E0004	.03	.8770
B:AGW	29540	1	2.9540E0004	.05	.8464
C:MGW	10641535	1	1.0642E0007	**17.91**	**.0515**
D:MP	7295	1	7.2948E0003	.01	.9229
E:AM	2089942	1	2.0899E0006	3.52	.2015
F:CP	264234	1	2.6423E0005	.44	.5798
G:SM	1837440	1	1.8374E0006	3.09	.2207
H:PDU	783885	1	7.8389E0005	1.32	.3695
I:HSC	690954	1	6.9095E0005	1.16	.3936
J:SC	866459	1	8.6646E0005	1.46	.3506
K:RA82	1014148	1	1.0141E0006	1.71	.3214
L:ADOC	39410348	1	3.9410E0007	**66.35**	**.0147**
M:NCC	304784103	1	3.0478E0008	**513.09**	**.0019**
Total error	1188030	2	5.9402E0005		
Total (corr.)	363626662	15			

R-squared = 0.996733 R-squared (adj. for d.f.) = 0.975496

Table 4.2: Full ANOVA Table

The ANOVA for the reduced model, containing only significant factors, is displayed in Table 4.3. Note the R-squared value here of .975 indicates we have a good fit to the data. Other indications of an acceptable model are displayed in the standard diagnostic plot of the residuals and in the plot of observed versus predicted values included in Appendix G.2. Except for two possible outliers, the residuals

display homoscedasticity. The plot of observed versus predicted values shows a clear linear trend. Thus, the model selected closely approximates the true response surface representing system downtime as a function of part redundancies.

It is surprising that such a good fit was found with a linear model when diminishing decreases in downtime from additional parts would be expected. Since the levels for eight of the components represent one part versus two, the middle of the design region really has no interpretation. Therefore, since only the extreme points have meaning, the regression equation really does reflect a linear relationship for these components. Possibly what is happening is that the effect of these eight components is masking the diminishing returns from the other five components. Additionally, because of consistently high estimates of model availability, the region modelled is likely representative of the high end of the downtime returns curve. If this is the case, this relatively small region of the curve could be reasonably estimated by a linear model.

ANOVA for Average Annual Downtime - 13 factor study

Effect	Sum of Squares	DF	Mean Sq.	F-Ratio	P-value
C:MGW	10,641,535	1	1.0642E0007	14.53	.0025
L:ADOC	39,410,348	1	3.9410E0007	53.80	.0000
M:NCC	304,784,103	1	3.0478E0008	416.06	.0000
Total error	8,790,675	12	7.3256E0005		
Total (corr.)	363,626,662	15			

R-squared = 0.975825 R-squared (adj. for d.f.) = 0.969781

Table 4.3: Reduced ANOVA Table--Significant Factors

The suggested regression model in the uncoded region, with coefficients rounded to whole numbers is:

Annual System Downtime= 49376 - 1631 (MGW) - 1046 (ADOC) - 4365 (NCC)

where	MGW	=	uncoded number of Mission Gateway Software parts
	ADOC	=	uncoded number of ADOC workstations
	NCC	=	uncoded number of NCC workstations

Setting all of the significant factors to their high levels yields a downtime prediction of 556 minutes. This is the only prediction that can be compared to previous results, since previous results were calculated for a fully redundant system. This average seems high when compared to the analytical estimate of 7.422 minutes; however, the model standard deviation, approximately 856 minutes, explains this difference and emphasizes the fact that this is truly a coarse model of the system. In order for substantiated conclusions to be drawn from this type of regression model, an estimate of mean square error (MSE) on the order of the variances calculated for the 10-run averages in the runtime analysis would be appropriate (200-400 minutes). Obtaining this level of MSE could involve reduction of the design region and/or addition of experimental runs.

The fact that all significant component coefficients are negative means that the greater the part redundancy in these components, the smaller the prediction of annual system downtime. The negative coefficients therefore suggest that maximum redundancy in these significant components produces the system configuration with the highest model availability.

The coefficient of -4365 on the NCC workstations suggests that one NCC workstation is worth a savings of 4365 minutes of downtime annually, or a savings of approximately 48 system critical failures. Since the design region restricts us to a maximum of 9 NCC workstations, the greatest possible reduction in annual downtime is 39,281 minutes. This reduction translates to a change in model availability of .075 which is definitely significant. The NCC coefficient can also be compared with the ADOC coefficient of -1046. Since the NCC coefficient is roughly 4 times larger than the ADOC coefficient, this suggests that addition of one workstation in the NCC is worth roughly 4 times as much as addition of one workstation in the ADOC in terms of critical system downtime. Similarly, in terms of critical system downtime one NCC workstation is worth approximately 2.5 Mission Gateway software parts. These types of comparisons, made based on a model with less variance, could provide valuable information to system designers determining the most critical areas to increase redundancy in the system.

The redundancy of the components not represented in the regression model does not have a significant effect on average annual downtime compared to the error inherent in the model. The command post (CP) software component can be used to illustrate how this insignificance translates to system downtime. The CP software has two parts, each with an estimated availability of .999755. On the average, one command post software element, functioning without backup, contributes 128 minutes to total system downtime annually. When two command post software elements function together in the system, the component availability is ont he order of

.9999999, translating to a contribution of .05 minutes of system downtime annually by the entire component. The regression analysis considered the increase from .05 to 128 minutes to be insignificant because the level of uncertainty in the model renders relatively small changes indistinguishable from random variations. A mean square error (MSE) of 732,560 can potentially mask a wide range of factors and therefore indicates the model should be refined before use in real time system analysis.

V. Conclusions and Research Recommendations

5.1 Conclusions

Construction of a prototype availability simulation model of Granite Sentry fulfilled the primary purpose of this research as outlined by AFOTEC. Currently, the model provides a number of system performance measures as a function of component MTBFs and MTTRs. This model could be extended to incorporate effects of other factors known to influence the system such as maintenance concepts, supply levels, and software failures induced by message receipt. Given these types of extensions, the simulation could become more representative of the actual system.

Analysis of failure data prior to model construction supported the generally accepted use of exponentially distributed failure rates and lognormally distributed repair times. While other distributions like the weibull or lognormal often provided a better fit to the TBF data, the exponential distribution was always adequate. Similarly, a lognormal distribution always provided an adequate fit to the TTR data, though the weibull distribution was often selected as providing the best fit.

Use of a Microsoft Windows version of SLAMSYSTEM made building the initial network model extremely efficient. Modelling becomes somewhat object-oriented with the pictorial diagramming available in this software. Less time is wasted on model coding details, allowing more time to focus on model intricacies, which is especially useful during the initial phase of model construction.

Results of the runtime analysis suggested at least 6 runs are needed to reach a stabilized estimate of both the mean and variance of a system performance measure. Additionally, a runtime estimate of 20 times longer than the longest MTBF appeared to produce reasonably efficient and accurate results. In general, both runtime and replications should be determined by the accuracy of results desired--the greater the accuracy, the greater the number of replications.

This research strongly supports the use of analytical modelling as a fundamental analysis tool. Specifically during this research the analytical model developed, though a very basic system representation, proved to be essential for simulation model validation. It provided sanity checks essential to development of a representative simulation model.

5.2 Areas for Further Research

Since this research developed a specific model for Granite Sentry, the need still exists to assist AFOTEC in developing a generalized method for availability simulation modelling. Developing general modelling "subroutines" used to represent a wide variety of system components and their failures would greatly facilitate construction of these availability simulation models. The code developed in this and Brown's thesis (5) would be a solid starting point for developing this kind of generalized or object-oriented modelling. A segmented approach as outlined in Chapter 3 would be a structured means of beginning this research. The scope of this task could be limited through selection of specific system types to be modelled.

If time had permitted, a continuous-time markov chain model of the system would also have been constructed during this research. This type of analytical approach offers extensive system information through state transition and limiting probabilities. This type of model would require the assumption that repair times are exponentially distributed which, from Granite Sentry TTR data, would not be an unreasonable assumption.

With a model as complex as Granite Sentry, the system would likely first be decomposed and balance equations solved at the component level. Model development would begin with determination of the states which, at the component level, could represent the number of failed parts in the component. Components with two parts could be in three possible states--0, 1, or 2. Once balance equations were solved to determine state transition probabilities for components, this information would be combined at the system level. System level states could be represented by a n-dimensional vector whose entries represent the state of each of n components in the system. Values in the vector could be 0-1 variables indicating whether the critical number of component parts were functioning. Results of this type of analysis could provide details of system performance at both the component and system level.

Appendix A

1992 Failure Summary Database

IR/JCN	Culprit	Failure start (min)	Dwn Time	Critical	Relevant	Failure End (min)
9102-060	air gw	720	25	0	1	745
9102-069	air gw	32400	0	0	1	32400
9102-074	air gw	33840	1	0	0	33841
9102-101	air gw	52560	0	0	1	52560
9102-108	air gw	65520	4	0	1	65524
9102-111	air gw	68400	4	0	1	68404
9102-124	air gw	81360	0	0	1	81360
2954003	air gw	425520	72	0	n/a	425592
9202-080	air gw	491760	96	0	n/a	491856
9202-019	air gw	415440	73	0	1	415513
ir-225	am	190800	5		1	190805
ir-232	am	199440	1		1	199441
ir-248	am	209520	62		1	209582
9202-035	am	425520	0	0	1	425520
9202-040	am	434160	12	1	1	434172
9102-112	am	69840	1	0	1	69841
9102-153	am	102960	216	1	1	103176
9102-196	am	150480	0	0	1	150480
9202-098	am	526320	0	0	1	526320
ir-238	cp	200880	60	0	1	200940
ir-074	cp	270000	110	0	1	270110
9201-116	cp	372240	0	0	n/a	372240
9202-021	cp	418320	17	1	1	418337
9202-047	cp	441360	2	0	1	441362
9202-049	cp	445680	1	1	1	445681
9202-058	cp	455760	36	1	1	455796
9202-068	cp	470160	2	1	1	470162
9102-061	cp	6480	0	0	1	6480
9102-065	cp	29520	0	0	1	29520
9202-002	gw	385200	6	1	n/a	385206
9202-025	gw	422640	2	0	n/a	422642
3284008	gw	471600	85	1	n/a	471685
9202-070	gw	473040	214	1	n/a	473254

9102-128	gw	82800	0	0	1	82800
ir-054	mp	245520	225	0	1	245745
ir-060	mp	245520	395		1	245915
9202-044	mp	439920	31	1	1	439951
304012	mp	42480	50	0	1	42530
304028	mp	42480	130	0	1	42610
9102-202	mp	159120	162	0	1	159282
924006	mp	131760	30	0	1	131790
9102-114	mp	72720	350	0	1	73070
3004001	mp	431280	106	0	1	431386
9202-078	mp	488880	0	0	0	488880
9102-073	mw gw	33840	4	0	1	33844
9102-123	mw gw	81360	15	0	1	81375
9102-156	mw gw	102960	23	0	1	102983
9102-158	mw gw	105840	4	0	1	105844
9102-164	mw gw	110160	11	1	1	110171
9102-179	mw gw	134640	1	0 n/a		134641
ir-242	mw gw	205200	30		0	205230
ir-247	mw gw	209520	2		1	209522
9201-088	mw gw	307440	81	0 n/a		307521
9201-089	mw gw	316080	92	0 n/a		316172
9202-024	mw gw	422640	26	0 n/a		422666
9202-038	mw gw	432720	1	1 n/a		432721
9202-051	mw gw	445680	0	0 n/a		445680
9202-054	mw gw	447120	11	1 n/a		447131
9202-056	mw gw	451440	2	1 n/a		451442
9202-062	mw gw	465840	3	1 n/a		465843
9202-065	mw gw	467280	96	1 n/a		467376
3264012	mw gw	468720	318	1 n/a		469038
9202-081	mw gw	491760	0	0 n/a		491760
9102-076	mw gw	41040	4	0	1	41044
9102-089	mw gw	46800	1	0	1	46801
9202-023	mw gw	418320	38	0 n/a		418358
2944010	mw gw	424080	215	0 n/a		424295
9202-029	mw gw	425520	10	0 n/a		425530
9202-077	mw gw	486000	19	0 n/a		486019
9202-077	mw gw	487440	31	0 n/a		487471
9102-191	ra82	144720	105	0	1	144825
1434013	ra82	205200	169		1	205369

ir-009	ra82	216720	255	0	1	216975
ir-015	ra82	219600	175	0	1	219775
ir-020	ra82	221040	191	0	1	221231
ir-026	ra82	223920	50	0	0	223970
ir-055	ra82	245520	165	0	1	245685
1774006	ra82	254160	110		1	254270
9201-090	ra82	323280	645	0	1	323925
9201-100	ra82	337680	65	1	1	337745
9201-101	ra82	342000	185	0	1	342185
9202-032	ra82	426960	443	0	1	427403
3134005	ra82	450000	23	0	1	450023
9202-082	ra82	494640	285	0	1	494925
9202-084	ra82	494640	455	0	1	495095
9202-083	ra82	494640	60	0	1	494700
9202-085	ra82	496080	76	0	1	496156
9202-096	ra82	524880	802	0	1	525682
9102-176	sm	127440	4320	0	0	131760
9102-187	sm	140400	45	0	1	140445
ir-234	sm	200880	150		1	201030
ir-239	sm	202320	77		1	202397
ir-255	sm	212400	46		1	212446
ir-004	sm	215280	1	0	1	215281
ir-007	sm	216720	75	0	1	216795
ir-037	sm	232560	20	0	1	232580
ir-059	sm	245520	15	0	1	245535
9202-005	sm	390960	1	0	1	390961
9202-016	sm	406800	25	0	1	406825
9202-060	sm	458640	16	0	n/a	458656
9202-061	sm	461520	2	0	1	461522
9202-063	sm	465840	0	0	1	465840
9202-071	sm	473040	124	1	n/a	473164
9102-062	sm	7920	81	0	1	8001
9102-193	sm	147600	25	0	1	147625
1526009	ws hw	219600	40	0	0	219640
1004006	ws hw	143280	12	0	1	143292
1574004	ws hw	225360	15	0	1	225375
9102-203	ws hw 1	160560	137	0	1	160697
9102-199	ws hw 10	156240	57	0	1	156297
3004011	ws hw 10	431280	13	0	1	431293

9102-106	ws hw 10	64080	135	0	1	64215
684002	ws hw 11	97200	60	0	1	97260
1576007	ws hw 11	225360	45	0	0	225405
54012	ws hw 11	6480	40	0	1	6520
2454012	ws hw 12	352080	75	0	1	352155
9102-161	ws hw 13	108720	94	0	1	108814
3074004	ws hw 14	441360	140	0	1	441500
1174003	ws hw 15	167760	20	0	1	167780
3394004	ws hw 15	487440	90	0	1	487530
9102-135	ws hw 18	88560	70	0	1	88630
9102-119	ws hw 19	79920	75	0	1	79995
9102-155	ws hw 19	102960	70	0	1	103030
864008	ws hw 2	123120	15	0	1	123135
9102-184	ws hw 2	138960	154	0	1	139114
1294004	ws hw 2	185040	40	0	1	185080
564011	ws hw 23	79920	20	0	1	79940
3364001	ws hw 24	481680	30	0	1	481710
1794007	ws hw 25	258480	43	0	1	258523
3594002	ws hw 25	517680	101	0	1	517781
294008	ws hw 25	41040	128	0	1	41168
3354007	ws hw 26	473040	0	0	1	473040
3364007	ws hw 26	481680	10	0	1	481690
866002	ws hw 27	123120	5	0	n/a	123125
9102-174	ws hw 28	126000	74	0	1	126074
9201-102	ws hw 28	342000	60	0	1	342060
9202-034	ws hw 28	429840	20	0	1	429860
64006	ws hw 29	7920	75	0	1	7995
324012	ws hw 29	45360	110	0	1	45470
2394010	ws hw 3	343440	15	0	1	343455
3044007	ws hw 3	438480	59	0	1	438539
9201-114	ws hw 32	366480	1805	0	1	368285
9202-092	ws hw 32	522000	39	0	1	522039
3630003	ws hw 32	523440	84	0	1	523524
9202-093	ws hw 33	522000	89	0	1	522089
636006	ws hw 4	90000	151	n/a	1	90151
9102-183	ws hw 4	138960	91	0	1	139051
9102-190	ws hw 4	143280	45	0	1	143325
1744005	ws hw 4	245520	35	0	1	245555
2764006	ws hw 4	396720	79	0	1	396799

9202-089	ws hw 4	514800	34	0	1	514834
3604005	ws hw 4	517680	55	0	1	517735
9102-086	ws hw 4	43920	35	0	1	43955
434004	ws hw 4	61200	35	0	1	61235
314004	ws hw 5	43920	23	0	1	43943
9102-085	ws hw 5	43920	74	0	1	43994
9102-090	ws hw 5	46800	350	0	1	47150
9102-186	ws hw 6	138960	197	0	1	139157
1526005	ws hw 6	218160	40	0	0	218200
2074006	ws hw 6	297360	10	0	0	297370
1134019	ws hw 8	162000	12	0 n/a		162012
1526006	ws hw 8	218160	45	0	0	218205
224016	ws hw 8	30960	42	0	1	31002
9102-066	ws hw 8	30960	152	0	1	31112
714006	ws hw 9	101520	30	0	1	101550
ir-211	ws hw 9	179280	233	0	1	179513
ir-211	ws hw 9	179280	233	0	1	179513
9102-100	ws sw	51120	23	0	1	51143
9102-105	ws sw	64080	15	0	1	64095
9102-121	ws sw 1	79920	50	0	1	79970
9102-149	ws sw 1	100080	15	0	1	100095
ir-210	ws sw 1	174960	12	0	1	174972
ir-002	ws sw 1	215280	241	0	1	215521
ir-022	ws sw 1	223920	10	0	1	223930
ir-028	ws sw 1	226800	12	0	1	226812
ir-046	ws sw 1	236880	12	0	1	236892
ir-051	ws sw 1	241200	231	0	1	241431
374013	ws sw 1	52560	10	0	1	52570
494011	ws sw 1	69840	33	0	1	69873
9102-147	ws sw 10	97200	32	0	1	97232
9102-177	ws sw 10	130320	40	0	1	130360
ir-218	ws sw 10	189360	25	0	1	189385
1684003	ws sw 10	241200	20	0	1	241220
54001	ws sw 10	6480	45	0	1	6525
9102-094	ws sw 10	48240	13	0		48253
9102-162	ws sw 11	110160	51	0	1	110211
ir-036	ws sw 11	231120	49	0	1	231169
64013	ws sw 11	7920		0	1	7920
9102-163	ws sw 12	110160	165	0	1	110325

ir-14	ws sw 12	219600	45	0	1	219645
9102-080	ws sw 12	41040	25	0	1	41065
9102-109	ws sw 12	66960	60	0	1	67020
9102-115	ws sw 12	75600	15	0	1	75615
9102-120	ws sw 13	79920	728	0	1	80648
1134012	ws sw 13	162000	25	0	1	162025
ir-215	ws sw 13	182160	26	0	0	182186
ir-027	ws sw 13	226800	15	0	1	226815
9102-071	ws sw 13	32400	20	0	1	32420
9102-072	ws sw 14	32400	20	0	1	32420
ir-216	ws sw 14	182160	40	0	0	182200
ir-050	ws sw 15	241200	12	0	1	241212
1684003	ws sw 15	241200	20	0	1	241220
274001	ws sw 15	38160	37	0	1	38197
9102-130	ws sw 16	84240	53	0	1	84293
9102-198	ws sw 16	151920	73	0	1	151993
9102-126	ws sw 17	81360	275	0	1	81635
9102-171	ws sw 17	123120	30	0	1	123150
ir-209	ws sw 17	174960	33	0	1	174993
104002	ws sw 17	13680	12	0	1	13692
9102-087	ws sw 17	45360	72	0	1	45432
9102-131	ws sw 18	85680	41	0	1	85721
ir-053	ws sw 18	245520	25	0	1	245545
ir-042	ws sw 19	234000	20	0	1	234020
9102-194	ws sw 2	147600	298	0	1	147898
ir-204	ws sw 2	169200	50	0	1	169250
ir-206	ws sw 2	172080	12	0	1	172092
ir-214	ws sw 2	180720	30	0	1	180750
ir-023	ws sw 2	223920	25	0	1	223945
1636006	ws sw 2	234000	30	0	1	234030
ir-047	ws sw 2	236880	12	0	1	236892
ir-048	ws sw 2	239760	12	0	1	239772
9102-093	ws sw 2	48240	12	0	1	48252
9102-144	ws sw 22	94320	177	0	1	94497
9102-122	ws sw 23	79920	210	0	1	80130
9102-195	ws sw 23	149040	115	0	1	149155
ir-207	ws sw 24	172080	32	0	1	172112
ir-213	ws sw 24	180720	45	0	1	180765
ir-221	ws sw 24	190800	20		1	190820

ir-222	ws sw 24	190800	65		1	190865
ir-024	ws sw 24	223920	22	0	1	223942
ir-032	ws sw 24	228240	10	0	1	228250
ir-041	ws sw 24	234000	18	0	1	234018
9102-078	ws sw 24	41040	25	0	1	41065
9102-132	ws sw 25	85680	50	0	1	85730
9102-137	ws sw 25	88560	23	0	1	88583
9102-150	ws sw 25	101520	50	0	1	101570
9102-159	ws sw 25	107280	36	0	1	107316
9102-189	ws sw 25	141840	60	0	1	141900
9102-200	ws sw 25	156240	40	0	1	156280
ir-018	ws sw 25	221040	12	0	1	221052
ir-025	ws sw 25	223920	15	0	1	223935
ir-040	ws sw 25	232560	6	0	1	232566
ir-043	ws sw 25	234000	39	0	1	234039
ir-045	ws sw 25	236880	35	0	1	236915
9102-077	ws sw 25	41040	147	0	1	41187
9102-096	ws sw 25	49680	2	0	1	49682
9102-099	ws sw 25	51120	29	0	1	51149
9102-103	ws sw 25	55440	0	0	1	55440
9102-166	ws sw 26	111600	30	0	1	111630
9102-092	ws sw 26	48240	90	0	1	48330
9102-107	ws sw 26	65520	17	0	1	65537
9102-175	ws sw 27	128880	60	0	1	128940
284010	ws sw 27	39600	35	0	1	39635
9102-117	ws sw 28	75600	15	0	1	75615
9102-125	ws sw 28	81360	20	0	1	81380
9102-133	ws sw 28	87120	40	0	1	87160
9102-197	ws sw 28	151920	23	0	1	151943
ir-205	ws sw 28	170640	15	0	1	170655
ir-208	ws sw 28	173520	37	0	1	173557
ir-011	ws sw 28	216720	11	0	1	216731
ir-030	ws sw 28	226800	288	0	1	227088
ir-038	ws sw 28	232560	0	0	1	232560
9102-095	ws sw 28	48240	0	0	1	48240
874015	ws sw 29	124560	16	0	1	124576
ir-217	ws sw 3	183600	106	0	1	183706
1614009	ws sw 3	231120	20	0	1	231140
9102-138	ws sw 4	90000	20	0	1	90020

9102-139	ws sw 4	90000	20	0	1	90020
9102-140	ws sw 4	91440	35	1	1	91475
9102-145	ws sw 4	94320	67	0	1	94387
9102-146	ws sw 4	94320	90	0	1	94410
9102-148	ws sw 4	97200	25	0	1	97225
9102-167	ws sw 4	113040	35	0	1	113075
9102-181	ws sw 4	134640	101	0	1	134741
9102-182	ws sw 4	136080	13	0	1	136093
9102-192	ws sw 4	146160	47	0	1	146207
ir-13	ws sw 4	218160	223	0	1	218383
ir-031	ws sw 4	228240	22	0	1	228262
1614010	ws sw 4	231120	20	0	1	231140
9102-079	ws sw 4	41040	25	0	1	41065
9102-102	ws sw 4	52560	136	0	1	52696
9102-136	ws sw 5	88560	44	0	1	88604
9102-157	ws sw 5	102960	10	0	1	102970
9102-160	ws sw 5	108720	20	0	1	108740
ir-219	ws sw 5	187920	496		1	188416
ir-223	ws sw 5	190800	162		1	190962
ir-008	ws sw 5	216720	18	0	1	216738
9102-064	ws sw 5	23760	15	0	1	23775
9102-116	ws sw 6	77040	0	0	1	77040
9102-118	ws sw 6	77040	45	0	1	77085
9102-134	ws sw 6	87120	536	0	1	87656
9102-143	ws sw 6	91440	506	1	1	91946
ir-029	ws sw 6	226800	12	0	1	226812
9102-068	ws sw 6	15120	14	0	1	15134
9102-151	ws sw 7	101520	74	0	1	101594
ir-212	ws sw 7	180720	428	0	1	181148
ir-039	ws sw 7	232560	19	0	1	232579
9102-110	ws sw 8	66960	240	0	1	67200
ir-033	ws sw 9	228240	75	0	1	228315

Appendix B.1

Time Between Failure Anaylsis--Bestfit Fitted Distribution Statistics

Air Gateway Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,1.75e+4)	5.582109e-6	2.0	0.222222	4.0	4.131658
Expon(1.75e+4)	**5.582109e-6**	**3.0**	**0.182179**	**2.0**	**0.346446**
Gamma(0.72,2.44e+4)	3.446969e-6	1.0	0.359468	5.0	5.563495
Lognormal(2.19e+4,5.31e+4)	1.012304e-5	5.0	0.202314	3.0	0.309219
Weibull(0.86,1.62e+4)	6.723445e-6	4.0	0.150171	1.0	0.220815

Command Post Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,4.70e+4)	3.058447e-6	2.0	0.3	5.0	8.28478
Expon(4.70e+4)	**3.058447e-6**	**3.0**	**0.214333**	**3.0**	**0.294565**
Gamma(0.76,6.16e+4)	1.280441e-6	1.0	0.3	4.0	8.653718
Lognormal(4.93e+4,8.14e+4)	5.106247e-6	5.0	0.138725	1.0	0.164007
Weibull(0.93,4.54e+4)	3.079157e-6	4.0	0.187344	2.0	0.242456

Mission Processor

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,6.09e+4)	2.773113e-6	4.0	0.25	5.0	4.666643
Expon(6.09e+4)	**2.773113e-6**	**5.0**	**0.232635**	**3.0**	**0.367737**
Gamma(1.21,5.02e+4)	2.248435e-6	2.0	0.25	4.0	4.569653
Lognormal(6.25e+4,6.36e+4)	1.678232e-6	1.0	0.168533	1.0	0.214114
Weibull(1.27,6.60e+4)	2.339584e-6	3.0	0.172211	2.0	0.268005

Air Mission Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,3.88e+4)	1.236785e-5	3.0	0.375	5.0	10.64352
Expon(3.88e+4)	**1.236785e-5**	**4.0**	**0.199263**	**1.0**	**0.394123**
Gamma(1.60,2.42e+4)	8.398407e-6	2.0	0.375	4.0	10.701525
Lognormal(4.09e+4,4.58e+4)	1.393445e-5	5.0	0.23828	3.0	0.523748
Weibull(1.33,4.22e+4)	7.904357e-6	1.0	0.23713	2.0	0.417398

Mission Gateway Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,1.89e+4)	1.823668e-5	4.0	0.260789	3.0	14.338534
Expon(1.89e+4)	**1.823668e-5**	**5.0**	**0.260789**	**4.0**	**2.516227**
Gamma(0.44,4.27e+4)	1.279896e-5	1.0	0.970524	5.0	33.0914
Lognormal(1.88e+4,4.23e+4)	1.767248e-5	3.0	0.128473	1.0	0.482263
Weibull(0.71,1.49e+4)	1.634464e-5	2.0	0.154853	2.0	0.762252

RA82 Disk Drives

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,2.50e+4)	1.755651e-5	3.0	0.333333	5.0	15.574513
Expon(2.50e+4)	**1.755651e-5**	**4.0**	**0.177302**	**3.0**	**0.61115**
Gamma(0.85,2.94e+4)	1.255554e-5	1.0	0.333333	4.0	15.295113
Lognormal(3.37e+4,9.61e+4)	2.245817e-5	5.0	0.127433	2.0	0.293634
Weibull(0.83,2.27e+4)	1.516755e-5	2.0	0.111507	1.0	0.223854

Status Monitor Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,2.75e+4)	9.254091e-6	3.0	0.328832	3.0	10.134119
Expon(2.75e+4)	**9.254091e-6**	**4.0**	**0.328832**	**4.0**	**2.047664**
Gamma(0.42,6.60e+4)	5.114883e-6	1.0	0.894433	5.0	14.101811
Lognormal(2.71e+4,6.24e+4)	1.26926e-5	5.0	0.157314	1.0	0.313377
Weibull(0.69,2.11e+4)	8.386492e-6	2.0	0.205015	2.0	0.660556

Workstation Hardware

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,1.34e+5)	5.168615e-6	3.0	0.321429	5.0	52.408995
Expon(1.34e+5)	**5.168615e-6**	**4.0**	**0.080305**	**1.0**	**0.514415**
Gamma(1.00,1.34e+5)	5.11276e-6	2.0	0.321429	4.0	52.396768
Lognormal(1.85e+5,4.61e+5)	5.803985e-6	5.0	0.169765	3.0	1.516917
Weibull(0.94,1.30e+5)	4.736511e-6	1.0	0.097693	2.0	0.411985

Workstation Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Erlang(1.00,4.44e+4)	7.773732e-6	4.0	0.297521	4.0	97.050801
Expon(4.44e+4)	**7.773732e-6**	**5.0**	**0.12772**	**3.0**	**1.704748**
Gamma(0.85,5.25e+4)	5.44793e-6	1.0	0.322314	5.0	116.039477
Lognormal(5.28e+4,1.11e+5)	7.35714e-6	3.0	0.121435	2.0	1.686172
Weibull(0.91,4.23e+4)	6.018648e-6	2.0	0.091874	1.0	0.832917

Appendix B.2

MTTR Analysis--BESTFIT Fitted Distribution Statistics

Air Gateway Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Expon(27.50)	0.028907	1.0	0.465185	2.0	10.876182
Weibull(0.52,14.34)	0.046498	2.0	0.3	1.0	8.526877
Lognormal(1.02e+9,4.82e+18)	**0.258347**	**3.0**	**0.529048**	**3.0**	**10.686175**

Command Post Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Expon(22.80)	0.01678	1.0	0.516127	2.0	11.517562
Weibull(0.51,10.56)	0.025182	2.0	0.3	1.0	9.195967
Lognormal(2.98e+8,5.35e+1[illegible])	**0.18507**	**3.0**	**0.519814**	**3.0**	**10.823189**

Mission Processor

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Expon(1.61e+2)	7.546136e-4	1.0	0.200297	2.0	1.223248
Weibull(0.79,1.44e+2)	1.119206e-3	2.0	0.167994	1.0	1.18008
Lognormal(4.88e+7,1.53e+14)	**0.024448**	**3.0**	**0.514479**	**3.0**	**3.019424**

Air Mission Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Expon(33.00)	0.017029	1.0	0.52675	3.0	15.457855
Weibull(0.42,8.46)	0.031602	2.0	0.333333	1.0	10.128038
Lognormal(5.34e+8,2.92e+18)	**0.123322**	**3.0**	**0.518915**	**2.0**	**11.876528**

Mission Gateway Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Expon(46.27)	4.289477e-4	1.0	0.340329	3.0	8.592077
Weibull(0.56,25.04)	6.785464e-3	2.0	0.122088	1.0	2.525068
Lognormal(1.84e+4,7.79e+7)	**0.038165**	**3.0**	**0.229653**	**2.0**	**3.318796**

RA82 Disk Drives

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Expon(3.52e+2)	2.200876e-3	1.0	0.117967	1.0	0.33531
Weibull(1.24,3.78e+2)	2.25773e-3	2.0	0.148042	3.0	0.313863
Lognormal(3.87e+2,5.04e+2)	**2.925234e-3**	**3.0**	**0.127683**	**2.0**	**0.299404**

Status Monitor Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Expon(43.94)	4.592955e-3	1.0	0.205531	2.0	1.500284
Weibull(0.78,38.74)	4.820171e-3	2.0	0.155907	1.0	0.965251
Lognormal(1.44e+4,2.38e+7)	**0.042251**	**3.0**	**0.300548**	**3.0**	**2.176512**

Workstation Hardware

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Weibull(1.04,95.68)	1.853674e-3	1.0	0.119126	2.0	0.844058
Expon(94.05)	1.938607e-3	2.0	0.115707	1.0	0.964128
Lognormal(5.97e+2,7.49e+3)	**0.01304**	**3.0**	**0.225565**	**3.0**	**6.472211**

Workstation Software

Function	Chi-Square	Rank	K-S Test	Rank	A-D Test
Weibull(0.80,61.32)	3.657444e-3	1.0	0.174032	1.0	6.390286
Expon(71.49)	4.049147e-3	2.0	0.242755	2.0	9.340476
Lognormal(1.53e+3,1.02e+5)	**0.016114**	**3.0**	**0.324028**	**3.0**	**16.761454**

Appendix C

SLAM II MODEL CODE

```
GEN,BAUER,GRANSEN,1/7/1994,2,N,N,Y/Y,N,Y/1,72;
LIMITS,2,6,400;
SEEDS,32849743(9);

INTLC,XX(1)=0,XX(2)=0;
INTLC,XX(20)=0,XX(21)=0,XX(22)=0,XX(23)=0;
INTLC,XX(31)=0,XX(32)=0,XX(33)=0,XX(34)=0;
INTLC,XX(55)=0,XX(54)=0,XX(53)=0,XX(52)=0,XX(51)=0,XX(50)=0;
INTLC,XX(56)=0,XX(57)=0,XX(58)=0,XX(59)=0;
INTLC,XX(65)=0,XX(64)=0,XX(63)=0,XX(62)=0,XX(61)=0,XX(60)=0;
INTLC,XX(66)=0,XX(67)=0,XX(68)=0,XX(69)=0,XX(70)=0,XX(71)=0;
INTLC,XX(9)=30000001,XX(10)=30000001,XX(11)=30000001;
INTLC,XX(12)=30000001,XX(13)=30000001;
INTLC,XX(72)=0,XX(73)=0,XX(74)=0,XX(75)=0,XX(76)=0;
INTLC,XX(77)=0,XX(78)=0,XX(79)=0,XX(80)=0,XX(81)=0;
INTLC,XX(85)=0,XX(82)=0,XX(83)=0,XX(84)=0;
INTLC,XX(100)=0,XX(102)=0,XX(104)=0,XX(106)=0,XX(108)=0,XX(110)=0;
INTLC,XX(112)=0,XX(114)=0,XX(116)=0,XX(118)=0,XX(120)=0,XX(122)=0;
INTLC,XX(124)=0;

;  System critical failure global variables
INTLC,XX(120)=0,XX(121)=0,XX(130)=0,XX(131)=0;

;  Parametric Analysis Input Variables
INTLC,XX(200)=182784,XX(201)=161;        Gateway hardware--MTBF, MTTR
INTLC,XX(202)=35000,XX(203)=28;          Air Gateway Software
INTLC,XX(204)=19124,XX(205)=46;          Mission Gateway Software
INTLC,XX(206)=182784,XX(207)=161;        Mission Processor Hardware
INTLC,XX(208)=77520,XX(209)=33;          Air Mission Software
INTLC,XX(210)=93986,XX(211)=23;          Command Post Software
INTLC,XX(212)=1058400,XX(213)=60;        Power Distribution Unit
INTLC,XX(214)=1200000,XX(215)=60;        HSC 70 Controller
INTLC,XX(216)=1051200,XX(217)=60;        Star Coupler
INTLC,XX(218)=250270,XX(219)=352;        RA 82 Disk Drives
INTLC,XX(220)=82818,XX(221)=94;          Workstation Hardware
INTLC,XX(222)=44360,XX(223)=72;          Workstation Software
INTLC,XX(224)=137685,XX(225)=44;         Status Monitor Software

INTLC,XX(140)=0,XX(141)=0,XX(142)=0;     GW Downtime Counters
INTLC,XX(143)=0,XX(144)=0,XX(145)=0;     AGW Downtime Counters
INTLC,XX(146)=0,XX(147)=0,XX(148)=0;     MGW Downtime Counters
INTLC,XX(149)=0,XX(150)=0,XX(151)=0;     MP Downtime Counters
INTLC,XX(152)=0,XX(153)=0,XX(154)=0;     MP Downtime Counters
INTLC,XX(155)=0,XX(156)=0,XX(157)=0;     AM Downtime Counters
INTLC,XX(158)=0,XX(159)=0,XX(160)=0;     CP Downtime Counters
INTLC,XX(161)=0,XX(162)=0,XX(163)=0;     SM Downtime Counters
INTLC,XX(164)=0,XX(165)=0,XX(166)=0;     SM Downtime Counters
INTLC,XX(167)=0;                         RA82 Downtime Counter
INTLC,XX(168)=0;                         ADOC Downtime Counter
INTLC,XX(169)=0;                         NCC Downtime Counter
INTLC,XX(170)=0,XX(171)=0,XX(172)=0;     PDU Downtime Counters
INTLC,XX(173)=0,XX(174)=0,XX(175)=0;     HSC Downtime Counters
INTLC,XX(176)=0,XX(177)=0,XX(178)=0;     SC Downtime Counters
```

```
INTLC,XX(199)=0;                        Upper Bound on System Downtime

TIMST,XX(50),# GW1 FAILURES;
TIMST,XX(51),# GW2 FAILURES;
TIMST,XX(57),# A1GW FAILURES;
TIMST,XX(58),# A2GW FAILURES;
TIMST,XX(59),# M1GW FAILURES;
TIMST,XX(60),# M2GW FAILURES;
TIMST,XX(52),# MP1 FAILURES;
TIMST,XX(53),# MP2 FAILURES;
TIMST,XX(54),# MP3 FAILURES;
TIMST,XX(55),# MP4 FAILURES;
TIMST,XX(56),# MP5 FAILURES;
TIMST,XX(61),# AM1 FAILURES;
TIMST,XX(62),# AM2 FAILURES;
TIMST,XX(63),# CP1 FAILURES;
TIMST,XX(64),# CP2 FAILURES;
TIMST,XX(72),# SM1 FAILURES;
TIMST,XX(73),# SM2 FAILURES;
TIMST,XX(74),# SM3 FAILURES;
TIMST,XX(75),# SM4 FAILURES;
TIMST,XX(76),# SM5 FAILURES;
TIMST,XX(71),# RA82 FAILURES;
TIMST,XX(31),# ADOC HW FAILS;
TIMST,XX(32),# ADOC SW FAILS;
TIMST,XX(33),# NCC HW FAILS;
TIMST,XX(34),# NCC SW FAILS;
TIMST,XX(65),# PDU1 FAILURES;
TIMST,XX(66),# PDU2 FAILURES;
TIMST,XX(67),# HSC1 FAILURES;
TIMST,XX(68),# HSC2 FAILURES;
TIMST,XX(69),# SC1 FAILURES;
TIMST,XX(70),# SC2 FAILURES;

TIMST,XX(1),# HARDWARE FAIL;
TIMST,XX(2),# SOFTWARE FAIL;
TIMST,XX(130),# SYS CRIT FAIL;

TIMST,XX(100),# GW CURR FAILS;
TIMST,XX(102),#AGW CURR FAILS;
TIMST,XX(104),#MGW CURR FAILS;
TIMST,XX(106),# MP CURR FAILS;
TIMST,XX(108),# AM CURR FAILS;
TIMST,XX(110),# CP CURR FAILS;
TIMST,XX(112),#PDU CURR FAILS;
TIMST,XX(114),#HSC CURR FAILS;
TIMST,XX(116),# SC CURR FAILS;
TIMST,XX(118),#RA82 CURR FAILS;
TIMST,XX(120),#ADOC CURR WSDN;
TIMST,XX(122),#NCC CURR WSDN;
TIMST,XX(124),# SM CURR FAILS;
TIMST,XX(131),#CURR SYS FAILS;

TIMST,XX(140),DOWNTIME GW1;
TIMST,XX(141),DOWNTIME GW2;
TIMST,XX(142),UP BD DT GW;
TIMST,XX(143),DOWNTIME AGW1;
TIMST,XX(144),DOWNTIME AGW2;
TIMST,XX(145),UP BD DT AGW;
TIMST,XX(146),DOWNTIME MGW1;
```

```
TIMST,XX(147),DOWNTIME MGW2;
TIMST,XX(148),UP BD DT MGW;
TIMST,XX(149),DOWNTIME MP1;
TIMST,XX(150),DOWNTIME MP2;
TIMST,XX(151),DOWNTIME MP3;
TIMST,XX(152),DOWNTIME MP4;
TIMST,XX(153),DOWNTIME MP5;
TIMST,XX(154),UP BD DT MP;
TIMST,XX(155),DOWNTIME AM1;
TIMST,XX(156),DOWNTIME AM2;
TIMST,XX(157),UP BD DT AM;
TIMST,XX(158),DOWNTIME CP1;
TIMST,XX(159),DOWNTIME CP2;
TIMST,XX(160),UP BD DT CP;
TIMST,XX(161),DOWNTIME SM1;
TIMST,XX(162),DOWNTIME SM2;
TIMST,XX(163),DOWNTIME SM3;
TIMST,XX(164),DOWNTIME SM4;
TIMST,XX(165),DOWNTIME SM5;
TIMST,XX(166),UP BD DT SM;
TIMST,XX(167),UP BD DT RA82;
TIMST,XX(168),UP BD DT ADO;
TIMST,XX(169),UP BD DT ACC;
TIMST,XX(170),DOWNTIME PDU1;
TIMST,XX(171),DOWNTIME PDU2;
TIMST,XX(172),UP BD DT PDU;
TIMST,XX(173),DOWNTIME HSC1;
TIMST,XX(174),DOWNTIME HSC2;
TIMST,XX(175),UP BD DT HSC;
TIMST,XX(176),DOWNTIME SC1;
TIMST,XX(177),DOWNTIME SC2;
TIMST,XX(178),UP BD DT SC;

TIMST,XX(199),UP BD DT SYS;

NETWORK;

      GATE/1,GW1F,,1;
      GATE/2,GW2F,,1;
      GATE/3,MP1F,,1;
      GATE/4,MP2F,,1;
      GATE/5,MP3F,,1;
      GATE/6,MP4F,,1;
      GATE/7,MP5F,,1;
      GATE/8,A1GWT,,1;
      GATE/9,A2GWT,,1;
      GATE/10,M1GWT,,1;
      GATE/11,M2GWT,,1;
      GATE/12,AM1T,,1;
      GATE/13,AM2T,,1;
      GATE/14,CP1T,,1;
      GATE/15,CP2T,,1;
      GATE/16,PDU1,,1;
      GATE/17,PDU2,,1;
      GATE/18,SC1,,1;
      GATE/19,SC2,,1;
      GATE/20,HSC1,,1;
      GATE/21,HSC2,,1;
      GATE/22,RA82,,1;
      GATE/23,ADOC,,1;
```

```
     GATE/24,NCC,,1;
     GATE/25,SM1T,,1;
     GATE/26,SM2T,,1;
     GATE/27,SM3T,,1;
     GATE/28,SM4T,,1;
     GATE/29,SM5T,,1;
     GATE/30,SM1F,,1;
     GATE/31,SM2F,,1;
     GATE/32,SM3F,,1;
     GATE/33,SM4F,,1;
     GATE/34,SM5F,,1;
     GATE/35,SYSCRT,,1;

;*********************************************************************
;*********************************************************************
;  Failure Simulation Section
;*********************************************************************
;*********************************************************************
;     Gateway Hardware Failures
;*********************************************************************

;     Segment 1

      CREATE,,,1,1,1;
      ACTIVITY;
G1    GOON,1;
      ACTIVITY,EXPON(XX(200));
      ASSIGN,XX(50)=XX(50)+1,1;           GW1 Failure counter
      ASSIGN,XX(1)=XX(1)+1,1;             Hardware Failure Counter
      ASSIGN,XX(100)=XX(100)+1,1;         Current GW Failure counter
      ASSIGN,ATRIB(3)=1,ATRIB(5)=1,1;     Marking attributes--GW1, GW
      ASSIGN,ATRIB(4)=0,ATRIB(1)=TNOW,
        ATRIB(6)=RLOGN(XX(201),50),1;     Setting failure duration
      ACTIVITY,,XX(100).EQ.2,GCRC;        Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CG1   CLOSE,GW1F,1;
      ACTIVITY/1,ATRIB(6);                Failure Duration
      OPEN,GW1F,1;
      ACTIVITY;
      ASSIGN,XX(100)=XX(100)-1,1;         Decrement curr compon. fail
      ASSIGN,XX(140)=XX(140)+ATRIB(6),2;  Increment part downtime
      ACTIVITY,,,GCRO;                    Send to Open SYSCRT gate
      ACTIVITY,,,G1;                      Recirculate entity--next fail

;     Segment 2

      CREATE,,,1,1,1;
      ACTIVITY;
G2    GOON,1;
      ACTIVITY,EXPON(XX(200));
      ASSIGN,XX(51)=XX(51)+1,1;           GW2 Failure counter
      ASSIGN,XX(1)=XX(1)+1,1;             Hardware Failure Counter
      ASSIGN,XX(100)=XX(100)+1,1;         Current GW Failure counter
      ASSIGN,ATRIB(3)=2,ATRIB(5)=1,1;     Marking attributes--GW2, GW
      ASSIGN,ATRIB(4)=0,ATRIB(1)=TNOW,
          ATRIB(6)=RLOGN(XX(201),50),1;   Setting failure duration
      ACTIVITY,,XX(100).EQ.2,GCRC;        Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
```

```
CG2     CLOSE,GW2F,1;
        ACTIVITY/2,ATRIB(6);                Failure Duration
        OPEN,GW2F,1;
        ACTIVITY;
        ASSIGN,XX(100)=XX(100)-1,1;         Decrement curr compon. fail
        ASSIGN,XX(141)=XX(141)+ATRIB(6),2;  Increment part downtime
        ACTIVITY,,,GCRO;                    Send to Open SYSCRT gate
        ACTIVITY,,,G2;                      Recirculate entity--next fail

GCRC    COLCT,ALL,# GW CRIT FAIL,,2;
        ACTIVITY,,,SYCC;
        ACTIVITY;
        ASSIGN,ATRIB(4)=4,1;                Mark entity as critical fail
        ACTIVITY,,ATRIB(3).EQ.1,CG1;        Send entity to close part gate
        ACTIVITY,,,CG2;

GCRO    GOON,1;
        ACTIVITY,,ATRIB(4).EQ.4,SYCO;
        ACTIVITY;
        COLCT,ALL,NONCRIT GW,,1;
        TERMINATE;

;**********************************************************************
;       Air Gateway Software Failures
;**********************************************************************

;       Segment 1

        CREATE,,,1,1,1;
        ACTIVITY;
A1GW    GOON,1;
        ACTIVITY,EXPON(XX(202));
        ASSIGN,XX(57)=XX(57)+1,1;           Part Failure counter
        ASSIGN,XX(2)=XX(2)+1,1;             Software Failure Counter
        ASSIGN,XX(102)=XX(102)+1,1;         Current GW Failure counter
        ASSIGN,ATRIB(3)=1,ATRIB(5)=2,1;     Marking attributes--AGW1, AGW
        ASSIGN,ATRIB(4)=0,ATRIB(1)=TNOW,
          ATRIB(6)=RLOGN(XX(203),50),1;     Setting failure duration
        ACTIVITY,,XX(102).EQ.2,ACRC;        Send to close SYSCRT gate
        ACTIVITY;
        GOON,1;
CA1G    CLOSE,A1GWT,1;
        ACTIVITY/3,ATRIB(6);                Failure Duration
        OPEN,A1GWT,1;
        ACTIVITY;
        ASSIGN,XX(102)=XX(102)-1,1;         Decrement curr compon. fail
        ASSIGN,XX(143)=XX(143)+ATRIB(6),2;  Increment part downtime
        ACTIVITY,,,ACRO;                    Send to Open SYSCRT gate
        ACTIVITY,,,A1GW;                    Recirculate entity--next fail

;       Segment 2

        CREATE,,,1,1,1;
        ACTIVITY;
A2GW    GOON,1;
        ACTIVITY,EXPON(XX(202));
        ASSIGN,XX(58)=XX(58)+1,1;           Part Failure counter
        ASSIGN,XX(2)=XX(2)+1,1;             Software Failure Counter
        ASSIGN,XX(102)=XX(102)+1,1;         Current GW Failure counter
        ASSIGN,ATRIB(3)=2,ATRIB(5)=2,1;     Marking attributes--AGW2, AGW
```

```
        ASSIGN,ATRIB(4)=0,ATRIB(1)=TNOW,
          ATRIB(6)=RLOGN(XX(203),50),1;      Setting failure duration
        ACTIVITY,,XX(102).EQ.2,ACRC;         Send to close SYSCRT gate
        ACTIVITY;
        GOON,1;
CA2G    CLOSE,A2GWT,1;
        ACTIVITY/4,ATRIB(6);                 Failure Duration
        OPEN,A2GWT,1;
        ACTIVITY;
        ASSIGN,XX(102)=XX(102)-1,1;          Decrement curr compon. fail
        ASSIGN,XX(144)=XX(144)+ATRIB(6),2;   Increment part downtime
        ACTIVITY,,,ACRO;                     Send to Open SYSCRT gate
        ACTIVITY,,,A2GW;                     Recirculate entity--next fail


ACRC    COLCT,ALL,# AGW CRIT FAIL,,2;
        ACTIVITY,,,SYCC;
        ACTIVITY;
        ASSIGN,ATRIB(4)=4,1;                 Mark entity as critical fail
        ACTIVITY,,ATRIB(3).EQ.1,CA1G;        Send entity to close part gate
        ACTIVITY,,,CA2G;

ACRO    GOON,1;
        ACTIVITY,,ATRIB(4).EQ.4,SYCO;
        ACTIVITY;
        COLCT,ALL,NONCRIT AGW,,1;
        TERMINATE;

;********************************************************************
;       Mission Gateway Software Failures
;********************************************************************

;       Segment 1

        CREATE,,,1,1,1;
        ACTIVITY;
M1GW    GOON,1;
        ACTIVITY,EXPON(XX(204));
        ASSIGN,XX(59)=XX(59)+1,1;            Part Failure counter
        ASSIGN,XX(2)=XX(2)+1,1;              Software Failure Counter
        ASSIGN,XX(104)=XX(104)+1,1;          Current GW Failure counter
        ASSIGN,ATRIB(3)=1,ATRIB(5)=3,1;      Marking attributes--MGW1, MGW
        ASSIGN,ATRIB(4)=0,
          ATRIB(6)=RLOGN(XX(205),50),1;      Setting failure duration
        ACTIVITY,,XX(104).EQ.2,MGCC;         Send to close SYSCRT gate
        ACTIVITY;
        GOON,1;
CM1G    CLOSE,M1GWT,1;
        ACTIVITY/5,ATRIB(6);                 Failure Duration
        OPEN,M1GWT,1;
        ACTIVITY;
        ASSIGN,XX(104)=XX(104)-1,1;          Decrement curr compon. fail
        ASSIGN,XX(146)=XX(146)+ATRIB(6),2;   Increment part downtime
        ACTIVITY,,,MGCO;                     Send to Open SYSCRT gate
        ACTIVITY,,,M1GW;                     Recirculate entity--next fail

;       Segment 2

        CREATE,,,1,1,1;
        ACTIVITY;
```

```
M2GW  GOON,1;
      ACTIVITY,EXPON(XX(204));
      ASSIGN,XX(60)=XX(60)+1,1;           Part Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;             Software Failure Counter
      ASSIGN,XX(104)=XX(104)+1,1;         Current GW Failure counter
      ASSIGN,ATRIB(3)=2,ATRIB(5)=3,1;     Marking attributes--MGW2, MGW
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(205),50),1;     Setting failure duration
      ACTIVITY,,XX(104).EQ.2,MGCC;        Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CM2G  CLOSE,M2GWT,1;
      ACTIVITY/6,ATRIB(6);                Failure Duration
      OPEN,M2GWT,1;
      ACTIVITY;
      ASSIGN,XX(104)=XX(104)-1,1;         Decrement curr compon. fail
      ASSIGN,XX(147)=XX(147)+ATRIB(6),2; Increment part downtime
      ACTIVITY,,,MGCO;                    Send to Open SYSCRT gate
      ACTIVITY,,,M2GW;                    Recirculate entity--next fail

MGCC  COLCT,ALL,# MGW CRIT FAIL,,2;
      ACTIVITY,,,SYCC;
      ACTIVITY;
      ASSIGN,ATRIB(4)=4,1;
      ACTIVITY,,ATRIB(3).EQ.1,CM1G;
      ACTIVITY,,,CM2G;

MGCO  GOON,1;
      ACTIVITY,,ATRIB(4).EQ.4,SYCO;
      ACTIVITY;
      COLCT,ALL,NONCRIT MGW,,1;
      TERMINATE;

;*******************************************************************
;     Mission Processor Hardware Failures
;*******************************************************************

;     Segment 1

      CREATE,,,1,1,1;
      ACTIVITY;
MP1   GOON,1;
      ACTIVITY,EXPON(XX(206));
      ASSIGN,XX(52)=XX(52)+1,1;           MP1 Failure counter
      ASSIGN,XX(1)=XX(1)+1,1;             Hardware Failure Counter
      ASSIGN,XX(106)=XX(106)+1,1;         Current MP failure counter
      ASSIGN,ATRIB(3)=1,ATRIB(5)=4,1;     Marking attributes--MP1,MP
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(207),50),1;     Setting failure duration
      ACTIVITY,,XX(106).EQ.5,MPCC;        Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CMP1  CLOSE,MP1F,1;
      ACTIVITY/7,ATRIB(6);                Failure Duration
      OPEN,MP1F,1;
      ACTIVITY;
      ASSIGN,XX(106)=XX(106)-1,1;         Decrement curr compon. fail
      ASSIGN,XX(149)=XX(149)+ATRIB(6),2; Increment part downtime
      ACTIVITY,,,MPCO;                    Send to Open SYSCRT gate
      ACTIVITY,,,MP1;                     Recirculate entity--next fail
```

```
;       Segment 2

        CREATE,,,1,1,1;
        ACTIVITY;
MP2     GOON,1;
        ACTIVITY,EXPON(XX(206));
        ASSIGN,XX(53)=XX(53)+1,1;            Part Failure counter
        ASSIGN,XX(1)=XX(1)+1,1;              Hardware Failure Counter
        ASSIGN,XX(106)=XX(106)+1,1;          Current MP failure counter
        ASSIGN,ATRIB(3)=2,ATRIB(5)=4,1;      Marking attributes--MP1,MP
        ASSIGN,ATRIB(4)=0,
          ATRIB(6)=RLOGN(XX(207),50),1;      Setting failure duration
        ACTIVITY,,XX(106).EQ.5,MPCC;         Send to close SYSCRT gate
        ACTIVITY;
        GOON,1;
CMP2    CLOSE,MP2F,1;
        ACTIVITY/8,ATRIB(6);                 Failure Duration
        OPEN,MP2F,1;
        ACTIVITY;
        ASSIGN,XX(106)=XX(106)-1,1;          Decrement curr compon. fail
        ASSIGN,XX(150)=XX(150)+ATRIB(6),2;   Increment part downtime
        ACTIVITY,,,MPCO;                     Send to Open SYSCRT gate
        ACTIVITY,,,MP2;                      Recirculate entity--next fail

;       Segment 3

        CREATE,,,1,1,1;
        ACTIVITY;
MP3     GOON,1;
        ACTIVITY,EXPON(XX(206));
        ASSIGN,XX(54)=XX(54)+1,1;            MP1 Failure counter
        ASSIGN,XX(1)=XX(1)+1,1;              Hardware Failure Counter
        ASSIGN,XX(106)=XX(106)+1,1;          Current MP failure counter
        ASSIGN,ATRIB(3)=3,ATRIB(5)=4,1;      Marking attributes--MP1,MP
        ASSIGN,ATRIB(4)=0,
          ATRIB(6)=RLOGN(XX(207),50),1;      Setting failure duration
        ACTIVITY,,XX(106).EQ.5,MPCC;         Send to close SYSCRT gate
        ACTIVITY;
        GOON,1;
CMP3    CLOSE,MP3F,1;
        ACTIVITY/9,ATRIB(6);                 Failure Duration
        OPEN,MP3F,1;
        ACTIVITY;
        ASSIGN,XX(106)=XX(106)-1,1;          Decrement curr compon. fail
        ASSIGN,XX(151)=XX(151)+ATRIB(6),2;   Increment part downtime
        ACTIVITY,,,MPCO;                     Send to Open SYSCRT gate
        ACTIVITY,,,MP3;                      Recirculate entity--next fail

;       Segment 4

        CREATE,,,1,1,1;
        ACTIVITY;
MP4     GOON,1;
        ACTIVITY,EXPON(XX(206));
        ASSIGN,XX(55)=XX(55)+1,1;            Part Failure counter
        ASSIGN,XX(1)=XX(1)+1,1;              Hardware Failure Counter
        ASSIGN,XX(106)=XX(106)+1,1;          Current MP failure counter
        ASSIGN,ATRIB(3)=4,ATRIB(5)=4,1;      Marking attributes--MP1,MP
        ASSIGN,ATRIB(4)=0,
          ATRIB(6)=RLOGN(XX(207),50),1;      Setting failure duration
```

```
      ACTIVITY,,XX(106).EQ.5,MPCC;        Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CMP4  CLOSE,MP4F,1;
      ACTIVITY/10,ATRIB(6);               Failure Duration
      OPEN,MP4F,1;
      ACTIVITY;
      ASSIGN,XX(106)=XX(106)-1,1;         Decrement curr compon. fail
      ASSIGN,XX(152)=XX(152)+ATRIB(6),2;  Increment part downtime
      ACTIVITY,,,MPCO;                    Send to Open SYSCRT gate
      ACTIVITY,,,MP4;                     Recirculate entity--next fail

;     Segment 5

      CREATE,,,1,1,1;
      ACTIVITY;
MP5   GOON,1;
      ACTIVITY,EXPON(XX(206));
      ASSIGN,XX(56)=XX(56)+1,1;           Part Failure counter
      ASSIGN,XX(1)=XX(1)+1,1;             Hardware Failure Counter
      ASSIGN,XX(106)=XX(106)+1,1;         Current MP failure counter
      ASSIGN,ATRIB(3)=5,ATRIB(5)=4,1;     Marking attributes--MP1,MP
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(207),50),1;     Setting failure duration
      ACTIVITY,,XX(106).EQ.5,MPCC;        Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CMP5  CLOSE,MP5F,1;
      ACTIVITY/11,ATRIB(6);               Failure Duration
      OPEN,MP5F,1;
      ACTIVITY;
      ASSIGN,XX(106)=XX(106)-1,1;         Decrement curr compon. fail
      ASSIGN,XX(153)=XX(153)+ATRIB(6),2;  Increment part downtime
      ACTIVITY,,,MPCO;                    Send to Open SYSCRT gate
      ACTIVITY,,,MP5;                     Recirculate entity--next fail

MPCC  COLCT,ALL,# MP CRIT FAIL,,2;
      ACTIVITY,,,SYCC;
      ACTIVITY;
      ASSIGN,ATRIB(4)=4,1;
      ACTIVITY,,ATRIB(3).EQ.1,CMP1;       Entity sent to close part gate
      ACTIVITY,,ATRIB(3).EQ.2,CMP2;
      ACTIVITY,,ATRIB(3).EQ.3,CMP3;
      ACTIVITY,,ATRIB(3).EQ.4,CMP4;
      ACTIVITY,,ATRIB(3).EQ.5,CMP5;

MPCO  GOON,1;
      ACTIVITY,,ATRIB(4).EQ.4,SYCO;
      ACTIVITY;
      COLCT,ALL,NONCRIT MP,,1;
      TERMINATE;

;*********************************************************************
;     Air Mission Software Failures
;*********************************************************************

;     Segment 1

      CREATE,,,1,1,1;
      ACTIVITY;
```

```
AM1   GOON,1;
      ACTIVITY,EXPON(XX(208));
      ASSIGN,XX(61)=XX(61)+1,1;           Part Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;             Software Failure Counter
      ASSIGN,XX(108)=XX(108)+1,1;         Current AM failure counter
      ASSIGN,ATRIB(3)=1,ATRIB(5)=5,1;     Marking attributes--AM1, AM
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(209),50),1;     Setting failure duration
      ACTIVITY,,XX(108).EQ.2,AMCC;        Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CAM1  CLOSE,AM1T,1;
      ACTIVITY/12,ATRIB(6);               Failure Duration
      OPEN,AM1T,1;
      ACTIVITY;
      ASSIGN,XX(108)=XX(108)-1,1;         Decrement curr compon. fail
      ASSIGN,XX(155)=XX(155)+ATRIB(6),2;  Increment part downtime
      ACTIVITY,,,AMCO;                    Send to Open SYSCRT gate
      ACTIVITY,,,AM1;                     Recirculate entity--next fail

;     Segment 2

      CREATE,,,1,1,1;
      ACTIVITY;
AM2   GOON,1;
      ACTIVITY,EXPON(XX(208));
      ASSIGN,XX(62)=XX(62)+1,1;           Part Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;             Software Failure Counter
      ASSIGN,XX(108)=XX(108)+1,1;         Current AM Failure counter
      ASSIGN,ATRIB(3)=2,ATRIB(5)=5,1;     Marking attributes--AM2, AM
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(209),50),1;     Setting failure duration
      ACTIVITY,,XX(108).EQ.2,AMCC;        Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CAM2  CLOSE,AM2T,1;
      ACTIVITY/13,ATRIB(6);               Failure Duration
      OPEN,AM2T,1;
      ACTIVITY;
      ASSIGN,XX(108)=XX(108)-1,1;         Decrement curr compon. fail
      ASSIGN,XX(156)=XX(156)+ATRIB(6),2;  Increment part downtime
      ACTIVITY,,,AMCO;                    Send to Open SYSCRT gate
      ACTIVITY,,,AM2;                     Recirculate entity--next fail

AMCC  COLCT,ALL,# AM CRIT FAIL,,2;
      ACTIVITY,,,SYCC;
      ACTIVITY;
      ASSIGN,ATRIB(4)=4,1;                Mark entity as critical fail
      ACTIVITY,,ATRIB(3).EQ.1,CAM1;       Send entity to close part gate
      ACTIVITY,,,CAM2;

AMCO  GOON,1;
      ACTIVITY,,ATRIB(4).EQ.4,SYCO;
      ACTIVITY;
      COLCT,ALL,NONCRIT AM,,1;
      TERMINATE;

;**********************************************************************
;     Command Post Software Failures
;**********************************************************************
```

```
;       Segment 1

        CREATE,,,1,1,1;
        ACTIVITY;
CP1     GOON,1;
        ACTIVITY,EXPON(XX(210));
        ASSIGN,XX(63)=XX(63)+1,1;            Part Failure counter
        ASSIGN,XX(2)=XX(2)+1,1;              Software Failure Counter
        ASSIGN,XX(110)=XX(110)+1,1;          Current CP Failure counter
        ASSIGN,ATRIB(3)=1,ATRIB(5)=6,1;      Marking attributes--CP1, CP
        ASSIGN,ATRIB(4)=0,
          ATRIB(6)=RLOGN(XX(211),50),1;      Setting failure duration
        ACTIVITY,,XX(110).EQ.2,CPCC;         Send to close SYSCRT gate
        ACTIVITY;
        GOON,1;
CCP1    CLOSE,CP1T,1;
        ACTIVITY/14,ATRIB(6);                Failure Duration
        OPEN,CP1T,1;
        ACTIVITY;
        ASSIGN,XX(110)=XX(110)-1,1;          Decrement curr compon. fail
        ASSIGN,XX(158)=XX(158)+ATRIB(6),2;   Increment part downtime
        ACTIVITY,,,CPCO;                     Send to Open SYSCRT gate
        ACTIVITY,,,CP1;                      Recirculate entity--next fail

;       Segment 2

        CREATE,,,1,1,1;
        ACTIVITY;
CP2     GOON,1;
        ACTIVITY,EXPON(XX(210));
        ASSIGN,XX(64)=XX(64)+1,1;            Part Failure counter
        ASSIGN,XX(2)=XX(2)+1,1;              Software Failure Counter
        ASSIGN,XX(110)=XX(110)+1,1;          Current CP Failure counter
        ASSIGN,ATRIB(3)=2,ATRIB(5)=6,1;      Marking attributes--CP2, CP
        ASSIGN,ATRIB(4)=0,
          ATRIB(6)=RLOGN(XX(211),50),1;      Setting failure duration
        ACTIVITY,,XX(110).EQ.2,CPCC;         Send to close SYSCRT gate
        ACTIVITY;
        GOON,1;
CCP2    CLOSE,CP2T,1;
        ACTIVITY/15,ATRIB(6);                Failure Duration
        OPEN,CP2T,1;
        ACTIVITY;
        ASSIGN,XX(110)=XX(110)-1,1;          Decrement curr compon. fail
        ASSIGN,XX(159)=XX(159)+ATRIB(6),2;   Increment part downtime
        ACTIVITY,,,CPCO;                     Send to Open SYSCRT gate
        ACTIVITY,,,CP2;                      Recirculate entity--next fail

CPCC    COLCT,ALL,# CP CRIT FAIL,,2;
        ACTIVITY,,,SYCC;
        ACTIVITY;
        ASSIGN,ATRIB(4)=4,1;
        ACTIVITY,,ATRIB(3).EQ.1,CCP1;
        ACTIVITY,,,CCP2;

CPCO    GOON,1;
        ACTIVITY,,ATRIB(4).EQ.4,SYCO;
        ACTIVITY;
        COLCT,ALL,NONCRIT CP,,1;
        TERMINATE;
```

```
;*************************************************************************
;     Status Monitor Software Failures
;*************************************************************************

;     Segment 1

      CREATE,,,1,1,1;
      ACTIVITY;
S1    GOON,1;
      ACTIVITY,EXPON(XX(224));
      ASSIGN,XX(72)=XX(72)+1,1;             Part Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;               Software Failure Counter
      ASSIGN,XX(124)=XX(124)+1,1;           Current SM failure counter
      ASSIGN,ATRIB(3)=1,ATRIB(5)=7,1;       Marking attributes--S1,SM
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(225),50),1;       Setting failure duration
      ACTIVITY,,XX(124).EQ.5,SMCC;          Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CSM1  CLOSE,SM1T,1;
      ACTIVITY/16,ATRIB(6);                  Failure Duration
      OPEN,SM1T,1;
      ACTIVITY;
      ASSIGN,XX(124)=XX(124)-1,1;           Decrement curr compon. fail
      ASSIGN,XX(161)=XX(161)+ATRIB(6),2;    Increment part downtime
      ACTIVITY,,,SMCO;                      Send to Open SYSCRT gate
      ACTIVITY,,,S1;                        Recirculate entity--next fail

;     Segment 2

      CREATE,,,1,1,1;
      ACTIVITY;
S2    GOON,1;
      ACTIVITY,EXPON(XX(224));
      ASSIGN,XX(73)=XX(73)+1,1;             Part Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;               Software Failure Counter
      ASSIGN,XX(124)=XX(124)+1,1;           Current SM failure counter
      ASSIGN,ATRIB(3)=2,ATRIB(5)=7,1;       Marking attributes--S2,SM
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(225),50),1;       Setting failure duration
      ACTIVITY,,XX(124).EQ.5,SMCC;          Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CSM2  CLOSE,SM2T,1;
      ACTIVITY/17,ATRIB(6);                 Failure Duration
      OPEN,SM2T,1;
      ACTIVITY;
      ASSIGN,XX(124)=XX(124)-1,1;           Decrement curr compon. fail
      ASSIGN,XX(162)=XX(162)+ATRIB(6),2;    Increment part downtime
      ACTIVITY,,,SMCO;                      Send to Open SYSCRT gate
      ACTIVITY,,,S2;                        Recirculate entity--next fail

;     Segment 3

      CREATE,,,1,1,1;
      ACTIVITY;
S3    GOON,1;
      ACTIVITY,EXPON(XX(224));
      ASSIGN,XX(74)=XX(74)+1,1;             Part Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;               Software Failure Counter
```

```
      ASSIGN,XX(124)=XX(124)+1,1;          Current SM failure counter
      ASSIGN,ATRIB(3)=3,ATRIB(5)=7,1;      Marking attributes--SM3,SM
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(225),50),1;      Setting failure duration
      ACTIVITY,,XX(124).EQ.5,SMCC;         Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CSM3  CLOSE,SM3T,1;
      ACTIVITY/18,ATRIB(6);                Failure Duration
      OPEN,SM3T,1;
      ACTIVITY;
      ASSIGN,XX(124)=XX(124)-1,1;          Decrement curr compon. fail
      ASSIGN,XX(163)=XX(163)+ATRIB(6),2;   Increment part downtime
      ACTIVITY,,,SMCO;                     Send to Open SYSCRT gate
      ACTIVITY,,,S3;                       Recirculate entity--next fail

;     Segment 4

      CREATE,,,1,1,1;
      ACTIVITY;
S4    GOON,1;
      ACTIVITY,EXPON(XX(224));
      ASSIGN,XX(75)=XX(75)+1,1;            Part Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;              Software Failure Counter
      ASSIGN,XX(124)=XX(124)+1,1;          Current SM failure counter
      ASSIGN,ATRIB(3)=4,ATRIB(5)=7,1;      Marking attributes--S4,SM
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(225),50),1;      Setting failure duration
      ACTIVITY,,XX(124).EQ.5,SMCC;         Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CSM4  CLOSE,SM4T,1;
      ACTIVITY/19,ATRIB(6);                Failure Duration
      OPEN,SM4T,1;
      ACTIVITY;
      ASSIGN,XX(124)=XX(124)-1,1;          Decrement curr compon. fail
      ASSIGN,XX(164)=XX(164)+ATRIB(6),2;   Increment part downtime
      ACTIVITY,,,SMCO;                     Send to Open SYSCRT gate
      ACTIVITY,,,S4;                       Recirculate entity--next fail

;     Segment 5

      CREATE,,,1,1,1;
      ACTIVITY;
S5    GOON,1;
      ACTIVITY,EXPON(XX(224));
      ASSIGN,XX(76)=XX(76)+1,1;            Part Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;              Software Failure Counter
      ASSIGN,XX(124)=XX(124)+1,1;          Current SM failure counter
      ASSIGN,ATRIB(3)=5,ATRIB(5)=7,1;      Marking attributes--S5,SM
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(225),50),1;      Setting failure duration
      ACTIVITY,,XX(124).EQ.5,SMCC;         Send to close SYSCRT gate
      ACTIVITY;
      GOON,1;
CSM5  CLOSE,SM5T,1;
      ACTIVITY/20,ATRIB(6);                Failure Duration
      OPEN,SM5T,1;
      ACTIVITY;
      ASSIGN,XX(124)=XX(124)-1,1;          Decrement curr compon. fail
```

```
       ASSIGN,XX(165)=XX(165)+ATRIB(6),2; Increment part downtime
       ACTIVITY,,,SMCO;                   Send to Open SYSCRT gate
       ACTIVITY,,,S5;                     Recirculate entity--next fail

SMCC   COLCT,ALL,# SM CRIT FAIL,,2;
       ACTIVITY,,,SYCC;
       ACTIVITY;
       ASSIGN,ATRIB(4)=4,1;
       ACTIVITY,,ATRIB(3).EQ.1,CSM1;      Entity sent to close part gate
       ACTIVITY,,ATRIB(3).EQ.2,CSM2;
       ACTIVITY,,ATRIB(3).EQ.3,CSM3;
       ACTIVITY,,ATRIB(3).EQ.4,CSM4;
       ACTIVITY,,ATRIB(3).EQ.5,CSM5;

SMCO   GOON,1;
       ACTIVITY,,ATRIB(4).EQ.4,SYCO;
       ACTIVITY;
       COLCT,ALL,NONCRIT SM,,1;
       TERMINATE;

;*********************************************************************
;      RA82 Disk Drive Failures
;*********************************************************************

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

       CREATE,,,1,1,1;
       ACTIVITY,,,R82;

R82    GOON,1;
       ACTIVITY,EXPON(XX(218));
       ASSIGN,XX(71)=XX(71)+1,1;          RA82 Total Failure counter
       ASSIGN,XX(1)=XX(1)+1,1;            Hardware Failure Counter
       ASSIGN,XX(118)=XX(118)+1,1;        Current RA82 Failure counter
       ASSIGN,ATRIB(5)=8,1;               Marking attribute--RA82 fail
```

```
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(219),50),1;    Setting failure duration
      ACTIVITY,,XX(118).GE.3,RAC;         Critical failure--close gates
      ACTIVITY;
R82A  GOON,1;
      ACTIVITY/21,ATRIB(6);               Failure Duration
      ASSIGN,XX(118)=XX(118)-1,1;         Decrement curr shadow set fail
      ACTIVITY,,,RAO;
      ACTIVITY,,,R82;                     Recirculate entity--next fail

RAC   COLCT,ALL,# RA82 CRIT FAIL,,2;
      ACTIVITY,,,SYCC;                    Send to close SYSCRT gate
      ACTIVITY;
      CLOSE,RA82,1;                       Close component gate
      ASSIGN,ATRIB(4)=4,1;                Mark entity as critical fail
      ACTIVITY,,,R82A;

RAO   GOON,2;
      ACTIVITY,,XX(118).LE.2.AND.ATRIB(4).EQ.4,SYCO;
      ACTIVITY,,XX(118).LE.2.AND.ATRIB(4).EQ.4,OPRA;
      ACTIVITY;
      COLCT,ALL,NONCRIT RA82,,1;
      TERMINATE;
OPRA  OPEN,RA82,1;
      TERMINATE;

;**********************************************************************
;     ADOC Workstation Failures
;**********************************************************************

;     ADOC Hardware Failures

      CREATE,,,1,1,1;
      ACTIVITY,,,AHW;

      CREATE,,,1,1,1;
      ACTIVITY,,,AHW;

      CREATE,,,1,1,1;
      ACTIVITY,,,AHW;

      CREATE,,,1,1,1;
      ACTIVITY,,,AHW;

      CREATE,,,1,1,1;
      ACTIVITY,,,AHW;

      CREATE,,,1,1,1;
      ACTIVITY,,,AHW;

AHW   GOON,1;
      ACTIVITY,EXPON(XX(220));
      ASSIGN,XX(31)=XX(31)+1,1;           ADOC Hardware Failure counter
      ASSIGN,XX(1)=XX(1)+1,1;             SYS Hardware Failure counter
      ASSIGN,XX(120)=XX(120)+1,1;         Current ADOC Failure counter
      ASSIGN,ATRIB(5)=9,ATRIB(3)=1,1;   Marking attributes--ADOC fail
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(221),50),1;     Setting failure duration
      ACTIVITY;
ADOF  GOON,1;
```

```
        ACTIVITY,,XX(120).GE.4,CADO;        Critical failure--close gates
        ACTIVITY;
ADOA    GOON,1;
        ACTIVITY/22,ATRIB(6);               Failure Duration
        ASSIGN,XX(120)=XX(120)-1,2;         Decrement curr WS down
        ACTIVITY,,,CADO;
        ACTIVITY,,ATRIB(3).EQ.1,AHW;        Recirc entity--next HW fail
        ACTIVITY,,ATRIB(3).EQ.2,ASW;        Recirc entity--next SW fail

CADC    COLCT,ALL,# ADOC CRITFAIL,,2;
        ACTIVITY,,,SYCC;                    Send to close SYSCRT gate
        ACTIVITY;
        CLOSE,ADOC,1;                       Close component gate
        ASSIGN,ATRIB(4)=4,1;                Mark entity as critical fail
        ACTIVITY,,,ADOA;

CADO    GOON,2;
        ACTIVITY,,XX(120).LE.3.AND.ATRIB(4).EQ.4,SYCO;
        ACTIVITY,,XX(120).LE.3.AND.ATRIB(4).EQ.4,OADO;
        ACTIVITY;
        COLCT,ALL,NONCRIT ADOC,,1;
        TERMINATE;
OADO    OPEN,ADOC,1;                        Re-open Component gate
        TERMINATE;

;       ADOC Software Failures

        CREATE,,,1,1,1;
        ACTIVITY,,,ASW;

        CREATE,,,1,1,1;
        ACTIVITY,,,ASW;

        CREATE,,,1,1,1;
        ACTIVITY,,,ASW;

        CREATE,,,1,1,1;
        ACTIVITY,,,ASW;

        CREATE,,,1,1,1;
        ACTIVITY,,,ASW;

        CREATE,,,1,1,1;
        ACTIVITY,,,ASW;


ASW     GOON,1;
        ACTIVITY,EXPON(XX(222));
        ASSIGN,XX(32)=XX(32)+1,1;           ADOC Software Failure counter
        ASSIGN,XX(2)=XX(2)+1,1;             SYS Software Failure counter
        ASSIGN,XX(120)=XX(120)+1,1;         Current ADOC Failure counter
        ASSIGN,ATRIB(5)=9,ATRIB(3)=2,1;     Marking attributes--ADOC fail
        ASSIGN,ATRIB(4)=0,
          ATRIB(6)=RLOGN(XX(223),50),1;     Setting failure duration
        ACTIVITY,,,ADOF;

;*********************************************************************
;       NCC Workstation Failures
;*********************************************************************
```

```
;       NCC Hardware Failures

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

        CREATE,,,1,1,1;
        ACTIVITY,,,NHW;

NHW     GOON,1;
        ACTIVITY,EXPON(XX(220));
        ASSIGN,XX(33)=XX(33)+1,1;          NCC Hardware Failure counter
        ASSIGN,XX(1)=XX(1)+1,1;            SYS Hardware Failure counter
        ASSIGN,XX(122)=XX(122)+1,1;        Current NCC Failure counter
        ASSIGN,ATRIB(5)=10,ATRIB(3)=1,1;   Marking attributes--NCC fail
        ASSIGN,ATRIB(4)=0,
          ATRIB(6)=RLOGN(XX(221),50),1;    Setting failure duration
        ACTIVITY;
NCCF    GOON,1;
        ACTIVITY,,XX(122).GE.3,CNCC;       Critical failure--close gates
        ACTIVITY;
NCCA    GOON,1;
        ACTIVITY/23,ATRIB(6);              Failure Duration
        ASSIGN,XX(122)=XX(122)-1,2;        Decrement curr WS down
        ACTIVITY,,,CNCO;
        ACTIVITY,,ATRIB(3).EQ.1,NHW;       Recirc entity--next HW fail
        ACTIVITY,,ATRIB(3).EQ.2,NSW;       Recirc entity--next SW fail

CNCC    COLCT,ALL,# NCC CRIT FAIL,,2;
        ACTIVITY,,,SYCC;                   Send to close SYSCRT gate
        ACTIVITY;
        CLOSE,NCC,1;                       Close component gate
        ASSIGN,ATRIB(4)=4,1;               Mark entity as critical fail
        ACTIVITY,,,NCCA;

CNCO    GOON,2;
        ACTIVITY,,XX(122).LE.2.AND.ATRIB(4).EQ.4,SYCO;
        ACTIVITY,,XX(122).LE.2.AND.ATRIB(4).EQ.4,ONCC;
        ACTIVITY;
        COLCT,ALL,NONCRIT NCC,,1;
```

```
      TERMINATE;
ONCC  OPEN,NCC,1;                          Re-open Component gate
      TERMINATE;

;     NCC Software Failures

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

      CREATE,,,1,1,1;
      ACTIVITY,,,NSW;

NSW   GOON,1;
      ACTIVITY,EXPON(XX(222));
      ASSIGN,XX(34)=XX(34)+1,1;            NCC Software Failure counter
      ASSIGN,XX(2)=XX(2)+1,1;              SYS Software Failure counter
      ASSIGN,XX(122)=XX(122)+1,1;          Current NCC Failure counter
      ASSIGN,ATRIB(5)=10,ATRIB(3)=2,1;     Marking attributes--NCC fail
      ASSIGN,ATRIB(4)=0,
        ATRIB(6)=RLOGN(XX(223),50),1;      Setting failure duration
      ACTIVITY,,,NCCF;

  ;*******************************************************************
;     System Critical Failure Summary Section
;*******************************************************************

SYCC  COLCT,ALL,SYS CRIT CLZ CT,,1;
      ACTIVITY;
      ASSIGN,XX(130)=XX(130)+1,XX(131)=XX(131)+1,1;      Counters
      ACTIVITY,,NNGAT(38).EQ.0,SCLZ;
      ACTIVITY,,,T1;
SCLZ  COLCT,ALL,ENTITYCLZGATE,,1;
      ACTIVITY;
      CLOSE,SYSCRT,1;
      ACTIVITY;
T1    TERMINATE;

SYCO  COLCT,ALL,SYS CRIT OPN CT,,1;
      ACTIVITY;
```

```
        ASSIGN,XX(131)=XX(131)-1,
              XX(199)=XX(199)+ATRIB(6),1;  Upper Bound SYS Downtime
        ACTIVITY,,XX(131).EQ.0,SOPN;
        ACTIVITY,,,T2;
SOPN    COLCT,ALL,ENTITYOPNGATE,,1;
        ACTIVITY;
        OPEN,SYSCRT,1;
        ACTIVITY;
T2      GOON,1;
        ACTIVITY,,ATRIB(5).EQ.1,XGW;
        ACTIVITY,,ATRIB(5).EQ.2,XAGW;
        ACTIVITY,,ATRIB(5).EQ.3,XMGW;
        ACTIVITY,,ATRIB(5).EQ.4,XMP;
        ACTIVITY,,ATRIB(5).EQ.5,XAM;
        ACTIVITY,,ATRIB(5).EQ.6,XCP;
        ACTIVITY,,ATRIB(5).EQ.7,XSM;
        ACTIVITY,,ATRIB(5).EQ.8,XRA8;
        ACTIVITY,,ATRIB(5).EQ.9,XADO;
        ACTIVITY,,ATRIB(5).EQ.10,XNCC;
        ACTIVITY,,ATRIB(5).EQ.11,XPDU;
        ACTIVITY,,ATRIB(5).EQ.12,XHSC;
        ACTIVITY,,ATRIB(5).EQ.12,XSC;

XGW     COLCT,ALL,GW SYCO,,1;
        ASSIGN,XX(142)=XX(142)+ATRIB(6),1;
        TERMINATE;
XAGW    COLCT,ALL,AGW SYCO,,1;
        ASSIGN,XX(145)=XX(145)+ATRIB(6),1;
        TERMINATE;
XMGW    COLCT,ALL,MGW SYCO,,1;
        ASSIGN,XX(148)=XX(148)+ATRIB(6),1;
        TERMINATE;
XMP     COLCT,ALL,MP SYCO,,1;
        ASSIGN,XX(154)=XX(154)+ATRIB(6),1;
        TERMINATE;
XAM     COLCT,ALL,AM SYCO,,1;
        ASSIGN,XX(157)=XX(157)+ATRIB(6),1;
        TERMINATE;
XCP     COLCT,ALL,CP SYCO,,1;
        ASSIGN,XX(160)=XX(160)+ATRIB(6),1;
        TERMINATE;
XSM     COLCT,ALL,SM SYCO,,1;
        ASSIGN,XX(166)=XX(166)+ATRIB(6),1;
        TERMINATE;
XRA8    COLCT,ALL,RA82 SYCO,,1;
        ASSIGN,XX(167)=XX(167)+ATRIB(6),1;
        TERMINATE;
XADO    COLCT,ALL,ADOC SYCO,,1;
        ASSIGN,XX(168)=XX(168)+ATRIB(6),1;
        TERMINATE;
XNCC    COLCT,ALL,NCC SYCO,,1;
        ASSIGN,XX(169)=XX(169)+ATRIB(6),1;
        TERMINATE;
XPDU    COLCT,ALL,PDU SYCO,,1;
        TERMINATE;
XHSC    COLCT,ALL,HSC SYCO,,1;
        TERMINATE;
XSC     COLCT,ALL,SC SYCO,,1;
        TERMINATE;
```

```
;*****************************************************************
;*****************************************************************
;  Granite Sentry Air and Missile Message Processing Section
;*****************************************************************
;*****************************************************************

;      Air and Missile Message Creation

AMSG   CREATE,UNFRM(5,10),,1,,1;
       ACTIVITY;
M1     ASSIGN,XX(1)=XX(1)+1,ATRIB(2)=1;
       ACTIVITY,,,GW;
MMSG   CREATE,RNORM(30,2),,1,,1;
       ACTIVITY;
M2     ASSIGN,XX(2)=XX(2)+1,ATRIB(2)=2;
       ACTIVITY,,,GW;

;*****************************************************************
;*****************************************************************
;      Gateway Hardware Segment

GW     GOON,1;
       ACTIVITY,,NNGAT(1).EQ.0,GW1;
       ACTIVITY,,NNGAT(2).EQ.0,GW2;
       ACTIVITY,,,MNRG;
GW1    COLCT,ALL,GW1 MSG;
       ACTIVITY,,,GWS;
GW2    COLCT,ALL,GW2 MSG;
       ACTIVITY,,,GWS;
MNRG   COLCT,ALL,MSG NR GATEWAYS;
       ACTIVITY,,,COUNT;

;*****************************************************************
;*****************************************************************
;      Gateway Software Segment

GWS    GOON,1;
       ACTIVITY,,ATRIB(2).EQ.1,AGW;
       ACTIVITY,,ATRIB(2).EQ.2,MGW;

;*****************************************************************
;      GW-Air Gateway Software--Air Messages

AGW    GOON,1;
       ACTIVITY,,NNGAT(8).EQ.0,A1R;
       ACTIVITY,,NNGAT(9).EQ.0,A2R;
       ACTIVITY,,,AGWN;
A1R    COLCT,ALL,A1GW MSG;
       ACTIVITY,,,MP;
A2R    COLCT,ALL,A2GW MSG;
       ACTIVITY,,,MP;
AGWNR COLCT,ALL,MSG NR AGW,1;
       ACTIVITY,,,COUNT;

;*****************************************************************
;      GW-Mission Gateway Software--Missile Messages

MGW    GOON,1;
       ACTIVITY,,NNGAT(10).EQ.0,MW1R;
       ACTIVITY,,NNGAT(11).EQ.0,MW2R;
```

```
       ACTIVITY,,,MGWN;
MW1R   COLCT,ALL,M1GW MSG;
       ACTIVITY,,,MP;
MW2R   COLCT,ALL,M2GW MSG;
       ACTIVITY,,,MP;
MGWNR  COLCT,ALL,MSG NR MGW;
       ACTIVITY,,,COUNT;

;***********************************************************************
;***********************************************************************
;      Mission Processors--Hardware

MP     GOON,1;
       ACTIVITY,,NNGAT(3).EQ.0,MP1R;
       ACTIVITY,,NNGAT(4).EQ.0,MP2R;
       ACTIVITY,,NNGAT(5).EQ.0,MP3R;
       ACTIVITY,,NNGAT(6).EQ.0,MP4R;
       ACTIVITY,,NNGAT(7).EQ.0,MP5R;
       ACTIVITY,,,MPNR;
MP1R   COLCT,ALL,MP1 MSG,,1;
       ACTIVITY,,,MPSW;
MP2R   COLCT,ALL,MP2 MSG,,1;
       ACTIVITY,,,MPSW;
MP3R   COLCT,ALL,MP3 MSG,,1;
       ACTIVITY,,,MPSW;
MP4R   COLCT,ALL,MP4 MSG,,1;
       ACTIVITY,,,MPSW;
MP5R   COLCT,ALL,MP5 MSG,,1;
       ACTIVITY,,,MPSW;
MPNR   COLCT,ALL,MSG NR M PROCESS;
       ACTIVITY,,,COUNT;

;***********************************************************************
;***********************************************************************
;      Mission Processor Software Segment

MPSW   GOON,1;
       ACTIVITY,,ATRIB(2).EQ.1,AMMP;
       ACTIVITY,,ATRIB(2).EQ.2,CPMP;

;***********************************************************************
;      MP-Air Mission Software--Air Messages

AMMP   GOON,1;
       ACTIVITY,,NNGAT(12).EQ.0,AM1R;
       ACTIVITY,,NNGAT(13).EQ.0,AM2R;
       ACTIVITY,,,AMNR;
AM1R   COLCT,ALL,AM1MP MSG;
       ACTIVITY,,,PDU;
AM2R   COLCT,ALL,AM2MP MSG;
       ACTIVITY,,,PDU;
AMNR   COLCT,ALL,MSG NR AMMP;
       ACTIVITY,,,COUNT;

;***********************************************************************
;      MP-Command Post Software--Missile Messages

CPMP   GOON,1;
       ACTIVITY,,NNGAT(14).EQ.0,CP1R;
       ACTIVITY,,NNGAT(15).EQ.0,CP2R;
```

```
      ACTIVITY,,,CPNR;
CP1R  COLCT,ALL,CP1MP MSG;
      ACTIVITY,,,PDU;
CP2R  COLCT,ALL,CP2MP MSG;
      ACTIVITY,,,PDU;
CPNR  COLCT,ALL,MSG NR CPMP;
      ACTIVITY,,,COUNT;

;*********************************************************************
;*********************************************************************
;  Power Distribution Unit Segment

PDU   GOON,1;
      ACTIVITY,,NNGAT(16).EQ.0,PD1;
      ACTIVITY,,NNGAT(17).EQ.0,PD2;
      ACTIVITY,,,PDUNR;
PD1   COLCT,ALL,PDU1 MSG;
      ACTIVITY,,,SC;
PD2   COLCT,ALL,PDU2 MSG;
      ACTIVITY,,,SC;
PDUNR COLCT,ALL,MSG NR PDU;
      ACTIVITY,,,COUNT;

;*********************************************************************
;*********************************************************************
;  Star Coupler Segment

SC    GOON,1;
      ACTIVITY,,NNGAT(18).EQ.0,SC1;
      ACTIVITY,,NNGAT(19).EQ.0,SC2;
      ACTIVITY,,,SCNR;
SC1   COLCT,ALL,SC1 MSG;
      ACTIVITY,,,HSC;
SC2   COLCT,ALL,SC2 MSG;
      ACTIVITY,,,HSC;
SCNR  COLCT,ALL,MSG NR SC;
      ACTIVITY,,,COUNT;

;*********************************************************************
;*********************************************************************
;  HSC 70 Segment

HSC   GOON,1;
      ACTIVITY,,NNGAT(20).EQ.0,H1;
      ACTIVITY,,NNGAT(21).EQ.0,H2;
      ACTIVITY,,,HSCNR;
H1    COLCT,ALL,HSC1 MSG;
      ACTIVITY,,,RA1;
H2    COLCT,ALL,HSC2 MSG;
      ACTIVITY,,,RA1;
HSCNR COLCT,ALL,MSG NR HSC;
      ACTIVITY,,,COUNT;

;*********************************************************************
;*********************************************************************
; RA82 Disk Drive Segment

RA1   GOON,1;
      ACTIVITY,,NNGAT(22).EQ.0,R1;
      ACTIVITY,,,RANR;
```

```
R1    COLCT,ALL,RA82 MSG;
      ACTIVITY,,,PRCES;
RANR  COLCT,ALL,MSG NR RA82;
      ACTIVITY,,,COUNT;

;*******************************************************************
;  Messages Received Count--total, air, and missile
;*******************************************************************

PRCES COLCT,BET,TOTAL MSG REC'D,,1;
      ACTIVITY,,ATRIB(2).EQ.1,AIRR;
      ACTIVITY,,ATRIB(2).EQ.2,MISS;
AIRR  COLCT,ALL,AIR REC'D;
      ACTIVITY,,,ADOC;
MISS  COLCT,ALL,MISSILE REC'D;
      ACTIVITY,,,ADOC;

;*******************************************************************
;  Messages Not Received Count--Total, Air, Missile
;*******************************************************************

COUNT COLCT,ALL,TOTAL MSG NR,1;
      ACTIVITY,,ATRIB(2).EQ.1,ANR;
      ACTIVITY,,ATRIB(2).EQ.2,MISN;
ANR   COLCT,ALL,AIR MSG NR;
      ACTIVITY,,,MNR;
MISNR COLCT,ALL,MISSILE MSG NR;
      ACTIVITY,,,MNR;
MNR   TERMINATE;

;*******************************************************************
;*******************************************************************
;  ADOC Workstations Segment

ADOC  GOON,1;
      ACTIVITY,,NNGAT(23).EQ.0,ADR;
      ACTIVITY,,,ADONR;
ADR   COLCT,ALL,ADOC MSG;
      ACTIVITY,,,NCC;
ADONR COLCT,ALL,MSG NR ADOC,1;
      ACTIVITY,,,NCC;

;*******************************************************************
;*******************************************************************
;  NCC Workstations Segment

NCC   GOON,1;
      ACTIVITY,,NNGAT(24).EQ.0,N1;
      ACTIVITY,,,NCCNR;
N1    COLCT,ALL,NCC MSG;
      ACTIVITY;
      TERM;
NCCNR COLCT,ALL,MSG NR NCC,1;
      ACTIVITY;
      TERM;

;*******************************************************************
;*******************************************************************
;     Granite Sentry Status Message Processing Section
;*******************************************************************
```

```
;*********************************************************************

;  Status Message Creation

      CREATE,5,,1,,1;
      ACTIVITY;
      ASSIGN,XX(85)=XX(85)+1;
      ACTIVITY,,,SMSW;

;*********************************************************************
;     Status Monitor Software Section
;*********************************************************************

SMSW  GOON,1;
      ACTIVITY,,NNGAT(25).EQ.0,SM1;
      ACTIVITY,,NNGAT(26).EQ.0,SM2;
      ACTIVITY,,NNGAT(27).EQ.0,SM3;
      ACTIVITY,,NNGAT(28).EQ.0,SM4;
      ACTIVITY,,NNGAT(29).EQ.0,SM5;
      ACTIVITY,,,SMNR;

SM1   GOON,1;
      ACTIVITY,,NNGAT(30).EQ.0,G1A;
      ACTIVITY,,NNGAT(26).EQ.0,SM2;
      ACTIVITY,,NNGAT(27).EQ.0,SM3;
      ACTIVITY,,NNGAT(28).EQ.0,SM4;
      ACTIVITY,,NNGAT(29).EQ.0,SM5;
      ACTIVITY,,,SMNR;

SM2   GOON,1;
      ACTIVITY,,NNGAT(31).EQ.0,G2A;
      ACTIVITY,,NNGAT(27).EQ.0,SM3;
      ACTIVITY,,NNGAT(28).EQ.0,SM4;
      ACTIVITY,,NNGAT(29).EQ.0,SM5;
      ACTIVITY,,,SMNR;

SM3   GOON,1;
      ACTIVITY,,NNGAT(32).EQ.0,G3A;
      ACTIVITY,,NNGAT(28).EQ.0,SM4;
      ACTIVITY,,NNGAT(29).EQ.0,SM5;
      ACTIVITY,,,SMNR;

SM4   GOON,1;
      ACTIVITY,,NNGAT(33).EQ.0,G4A;
      ACTIVITY,,NNGAT(29).EQ.0,SM5;
      ACTIVITY,,,SMNR;

SM5   GOON,1;
      ACTIVITY,,NNGAT(34).EQ.0,G5A;
      ACTIVITY,,,SMNR;


G1A   GOON,1;
      ACTIVITY,,ATRIB(1).GE.XX(9),G1B;
      ACTIVITY,,,SM1M;

G2A   GOON,1;
      ACTIVITY,,ATRIB(1).GE.XX(10),G2B;
      ACTIVITY,,,SM2M;
```

```
G3A     GOON,1;
        ACTIVITY,,ATRIB(1).GE.XX(11),G3B;
        ACTIVITY,,,SM3M;

G4A     GOON,1;
        ACTIVITY,,ATRIB(1).GE.XX(12),G4B;
        ACTIVITY,,,SM4M;

G5A     GOON,1;
        ACTIVITY,,ATRIB(1).GE.XX(13),G5B;
        ACTIVITY,,,SM5M;

;*********************************************************************
;       Status Monitor Software Message-Based Failures
;*********************************************************************

;       Message Based Software Failures--SM1 Segment

G1B     GOON,2;
        ACTIVITY;
        ACTIVITY,,,SMSW;
        ASSIGN,XX(77)=XX(77)+1,XX(9)=ATRIB(1)+50,1;
        CLOSE,SM1F,1;
        ACTIVITY/24,RLOGN(50,50);
        OPEN,SM1F,1;
        TERMINATE;

;       Message Based Software Failures--SM2 Segment

G2B     GOON,2;
        ACTIVITY;
        ACTIVITY,,,SMSW;
        ASSIGN,XX(78)=XX(78)+1,XX(10)=ATRIB(1)+50,1;
        CLOSE,SM2F,1;
        ACTIVITY/25,RLOGN(50,50);
        OPEN,SM2F,1;
        TERMINATE;

;       Message Based Software Failures--SM3 Segment

G3B     GOON,2;
        ACTIVITY;
        ACTIVITY,,,SMSW;
        ASSIGN,XX(79)=XX(79)+1,XX(11)=ATRIB(1)+50,1;
        ACTIVITY;
        CLOSE,SM3F,1;
        ACTIVITY/26,RLOGN(50,50);
        OPEN,SM3F,1;
        ACTIVITY;
        TERMINATE;

;       Message Based Software Failures--SM4 Segment

G4B     GOON,2;
        ACTIVITY;
        ACTIVITY,,,SMSW;
        ASSIGN,XX(80)=XX(80)+1,XX(12)=ATRIB(1)+60,1;
        ACTIVITY;
        CLOSE,SM4F,1;
        ACTIVITY/27,RLOGN(50,50);
```

```
       OPEN,SM4F,1;
       ACTIVITY;
       TERMINATE;

;      Message Based Software Failures--SM5 Segment

G5B    GOON,2;
       ACTIVITY;
       ACTIVITY,,,SMSW;
       ASSIGN,XX(81)=XX(81)+1,XX(13)=ATRIB(1)+60,1;
       ACTIVITY;
       CLOSE,SM5F,1;
       ACTIVITY/28,RLOGN(50,50);
       OPEN,SM5F,1;
       ACTIVITY;
       TERMINATE;

SM1M   COLCT,ALL,SM1 MSG,,1;
       ACTIVITY,,,SMR;

SM2M   COLCT,ALL,SM2 MSG,,1;
       ACTIVITY,,,SMR;

SM3M   COLCT,ALL,SM3 MSG,,1;
       ACTIVITY,,,SMR;

SM4M   COLCT,ALL,SM4 MSG,,1;
       ACTIVITY,,,SMR;

SM5M   COLCT,ALL,SM5 MSG,,1;
       ACTIVITY,,,SMR;


SMR    COLCT,ALL,STATUS MSG REC'D,,1;
       ACTIVITY;
       TERMINATE;

SMNR   COLCT,ALL,STATUS MSG NR;
       ACTIVITY;
       TERMINATE;

;*********************************************************************

       END;

INITIALIZE,,200000,Y;
FIN;
```

Appendix D

GRANITE SENTRY — Spreadsheet Availability Model

DEFINITIONS

indicates software (SW) component

indicates hardware (HW) component

f i = rate at which component i fails = 1/MTBF
ri = rate at which component i is repaired = 1/MTTR
Ai(t) = availability of component i
= probability component i is working at time t

ACRONYMS:
PDU = Power Distribution Unit
GW = Gateway (DEC VAX 8550)
A GW = Air Gateway Software
MW GW = Mission Gateway Software
SC = Star Coupler
HSC 70 = Disk Controllers
RA 82 = Disk drives (paired in 10 shadow sets)
MP = Mission Processors (DEC VAX 85500)
AM = Air Mission Software
CP = Command Post Software
SM = Status Monitor software
ADOC - Air Defense Operations Center Workstations (HW and SW)
NCC = NORAD Command Center Workstations (HW and SW)
FORMULAS: (Ref pp 440-445 of Ross's 4th Ed "Intro to Probability Models")

Availability of each part (eg. PDU1, PDU2, MP1, etc.) — Note: part = each block or circle (ie PDU1 block)

$$Ai(t) = (fi / fi + ri)$$

Availability of Parallel System (each separate component group, eg. PDU, GW, SC)

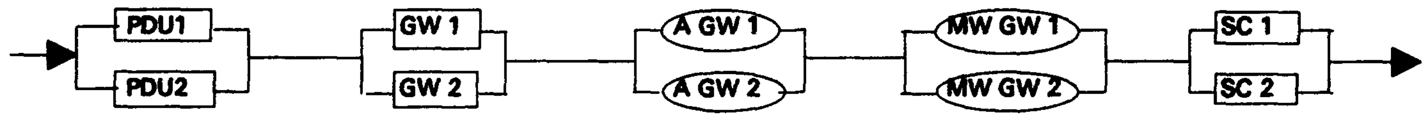
$$A(t) = 1 - ((r1/f1 + r1) * (r2/f2 + f2)* \dots)$$

Model Availability of entire Series System

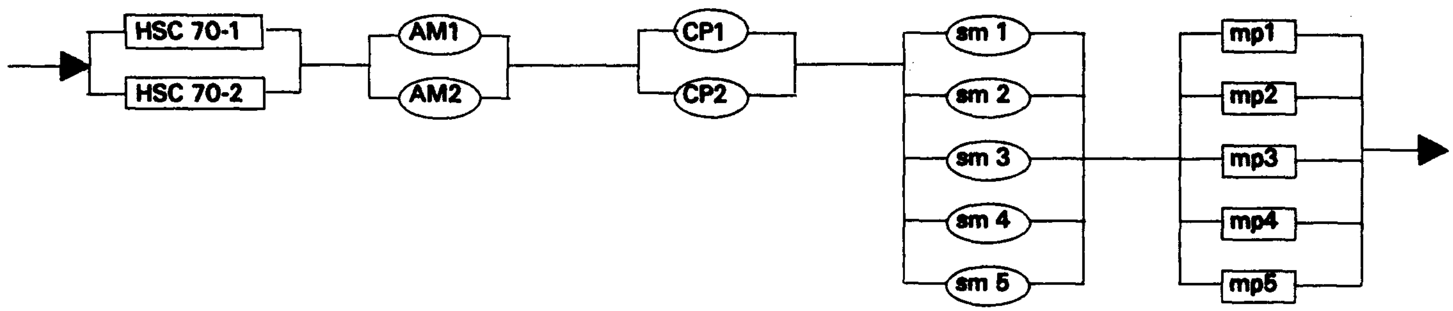
$$A(t) = A1 * A2 * A3 * \dots * A13$$

Note: There are 13 critical components in the model

GRANITE SENTRY AVAILABILITY MODEL



MTBF(1/fi)	1058400	182784	35000	19124	1000000
MTTR(1/ri)	60	161	28	46	60
Segment Availability	0.999943	0.99912	0.999201	0.9976	0.99994
Component Availability	1	0.999999	0.999999	0.999994	1



MTBF	1200000	77520	93986	137685	182784
MTTR	60	33	23	44	161
Segment Availability	0.99995	0.999574	0.999755	0.999681	0.99912
Component Avail.	1	1	1	1	1

	RA82	ADOC ws		NCC ws	
		HW	SW	HW	SW
MTBF	250270	82818	44360	82818	44360
MTTR	352	94	72	94	72
Segment Availability	0.999998	0.997734		0.997734	
String	(8 of 10 sets)	(3 of 6 ws)		(7 of 9 ws)	
Availability	0.999984	0.993217		0.984244	
	45 strings	20 strings		36 strings	
Component Avail.	1	1		1	

MODEL INTRINSIC AVAILABILITY = 0.999993

Appendix E

95% CONFIDENCE INTERVAL LIMITS FOR COMPONENT MTBFs

COMPONENT NAME	# in sys	Critical Number	# failures	Failures based on whole/seg	MTBF WHOLE	MTBF SEGMNT	Expon 95% Low	Expon 95% Up
Power Distribution Unit	2	1	none	source:	DEC	1058400		
Gateway (DEC VAX 8550)	2	1	4	from mp		304740	114372.304	811966.395
					MM est.	182784	91408.6838	365501.277
Air Gateway Software	2	1	10	whole	17500	35000	18831.7004	65049.888
Mission Gateway Software	2	1	26	whole	9562	19124	13020.9092	28087.6988
Star Couplers	2	1	none	unknown	at least 2yr	1051200		
HSC 70 Disk Controllers	2	1	1	source:	DEC	1200000		
RA82 disk drives	20	8/10 sets	16	whole	25027	500540	306644.015	817039.561
(10 shadow sets of 2)					slam input	250270	153322.008	408519.78
Mission Processors	5	1	8	whole	60948	304740	152397.815	609368.759
					MM est.	182784	91408.6838	365501.277
Air Mission Software	2	1	9	whole	38760	77520	40334.3166	148988.527
Command Post Software	2	1	10	whole	46993	93986	50569.034	174679.393
Status Monitor Software	5	1	17	whole	27537	137685	85592.5839	221481.329
Workstation hardware	15	10	59	segment	n/a	133701	103589.471	172565.389
					MM est.	82818	64166.1083	106891.649
Workstation software	15	10	130	segment	n/a	44360	37353.8054	52680.298

Appendix F.1

System Avg Downtime Analysis--per 1,000,000 minutes

Spreadsheet model with same center pts suggests avg system downtime per 1,000,000 minutes = 7.422

RUNTIME (millions of minutes)

	Run #	2	5	10	20	30
	1	0	1.554	2.94	21.5665	18.0847
	2	0	6.082	49.072	53.3825	23.543
	3	8.935	10.07	9.86	6.04	30.40933
RAW	4	68.315	4.63	56.498	33.287	16.70667
DATA	5	9.2	21.344	19.5	49.057	18.318
	6	34.965	6.126	11.784	22.6215	21.1813
	7	0	66.946	35.643	40.156	28.5043
	8	5.625	32.552	9.793	11.83	15.8687
	9	0	21.726	28.244	26.375	18.3123
	10	0	22.028	16.717	23.237	13.31767

MEANS

	1	0	0.777	1.47	10.78325	9.04235
	2	0	3.818	26.006	37.4745	20.81385
	3	2.97833333	5.902	20.624	26.9963333	24.0123433
#	4	19.3125	5.584	29.5925	28.569	22.185925
Runs	5	17.29	8.736	27.574	32.6666	21.41234
Averaged	6	20.2358333	8.301	24.9423333	30.9924167	21.3738333
	7	17.345	16.6788571	26.471	32.3015	22.3924714
	8	15.88	18.663	24.38625	29.7425625	21.577
	9	14.1155556	19.0033333	24.8148889	29.3683889	21.2142556
	10	12.704	19.3058	24.0051	28.75525	20.424597

VARIANCES

		2	5	10	20	30
	1	0	0	0	0	0
	2	0	10.251392	1064.08071	506.128928	14.8965194
	3	26.6114083	18.154864	618.938128	582.440394	38.1393385
#	4	1084.96094	12.5077387	734.361388	398.186718	38.7694414
Runs	5	834.173238	59.056324	571.142752	382.591667	32.0692498
Averaged	6	719.406194	48.3804092	498.468218	322.890673	25.6642964
	7	658.003583	531.63644	431.747934	281.071455	28.6502787
	8	581.172871	487.18296	404.839119	293.303679	29.8773309
	9	536.54564	427.327531	355.88781	257.900772	27.3269164
	10	496.854349	380.761555	322.902301	233.004524	30.5261987

Appendix F.2

System Avg Critical Failure Analysis--per 1,000,000 minutes

Spreadsheet model with same center pts suggests
avg # system failures per 1,000,000 minutes = .080806

RUNTIME (millions of minutes)

	Run #	2	5	10	20	30
	1	0	0.2	0.3	0.35	0.3333
	2	0	0.4	0.8	0.6	0.4
	3	0.5	0.2	0.5	0.25	0.3667
RAW	4	0.5	0.4	0.8	0.45	0.4333
DATA	5	0.5	0.4	0.5	0.45	0.4667
	6	0.5	0.2	0.3	0.35	0.4
	7	0	0.6	0.4	0.6	0.4
	8	0.5	0.6	0.4	0.35	0.3
	9	0	0.4	0.7	0.45	0.4
	10	0	0.2	0.2	0.4	0.3

MEANS

	1	0	0.1	0.15	0.175	0.16665
	2	0	0.3	0.55	0.475	0.36665
#	3	0.16666667	0.26666667	0.53333333	0.4	0.36666667
Runs	4	0.25	0.3	0.6	0.4125	0.383325
Averaged	5	0.3	0.32	0.58	0.42	0.4
	6	0.33333333	0.3	0.53333333	0.40833333	0.4
	7	0.28571429	0.34285714	0.51428571	0.43571429	0.4
	8	0.3125	0.375	0.5	0.425	0.3875
	9	0.27777778	0.37777778	0.52222222	0.42777778	0.38888889
	10	0.25	0.36	0.49	0.425	0.38

VARIANCES

		2	5	10	20	30
	1	0	0	0	0	0
	2	0	0.02	0.125	0.03125	0.00222444
#	3	0.08333333	0.01333333	0.06333333	0.0325	0.00111222
Runs	4	0.08333333	0.01333333	0.06	0.02229167	0.00185148
Averaged	5	0.075	0.012	0.047	0.017	0.00277889
	6	0.06666667	0.012	0.05066667	0.01441667	0.00222311
	7	0.07142857	0.02285714	0.0447619	0.0172619	0.00185259
	8	0.06696429	0.02785714	0.04	0.01571429	0.00283794
	9	0.06944444	0.02444444	0.03944444	0.01381944	0.00250056
	10	0.06944444	0.02488889	0.04544444	0.01236111	0.00301284

Appendix F.3

GRANITE SENTRY **Spreadsheet Availability Model** **Runtime Comparison Statistics**

# COMP	2	2	2	2	2
	PDU	GW	AGW	MGW	SC
MTBF(1/fi)	1058400	182784	35000	19124	1000000
MTTR(1/ri)	60	161	28	46	60
Segment Availability	0.99994331	0.99911995	0.99920064	0.99760042	0.99994
Component Availability	1	0.99999923	0.99999936	0.99999424	1

#COMP	2	2	2	5	5
	HSC	AM	CP	SM	MP
MTBF	1200000	77520	93986	137685	182784
MTTR	60	33	23	44	161
Segment Availability	0.99995	0.99957448	0.99975534	0.99968053	0.99911995
Component Avail.	1	0.99999982	0.99999994	1	1

#COMP	10	6		9	
	RA82	ADOC ws		NCC ws	
		HW	SW	HW	SW
MTBF	500540	82818	44360	82818	44360
MTTR	352	94	72	94	72
Segment Availability	0.99999951	0.99773382		0.99773382	
String	(8 of 10)	(3 of 6 ws)		(7 of 9 ws)	
Availability	0.99999605	0.99321686		0.98424419	
	45 strings	20 strings		36 strings	
Component Avail.	1	1		1	

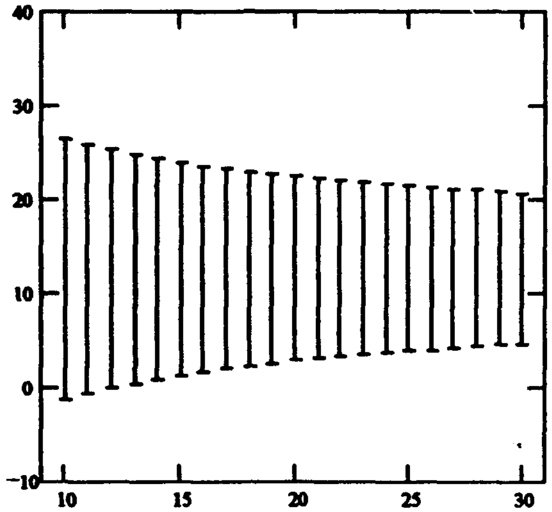
Avg # sys failures per 1,000,000 minutes	0.08080557	
Avg Sys Repair time	91.8461219	91.8461538
Avg Sys Downtime per 1,000,000 minutes	7.42167794	
SYSTEM AVAILABILITY = equals .999992578		0.99999258

Appendix F.4

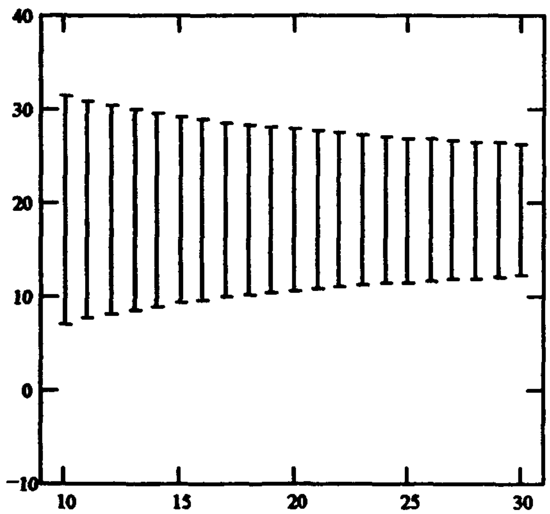
95% Confidence Intervals--
using 10-run averages to calculate the standard error of the mean

Y-axes reflect mean downtimes
X-axes reflect number of runs averaged

Runtime Group: 2 million minutes

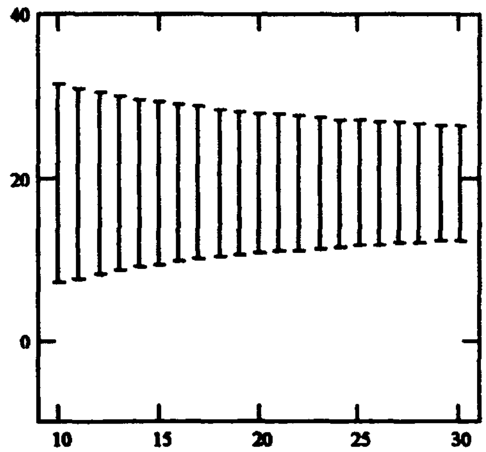


Runtime Group: 5 million minutes

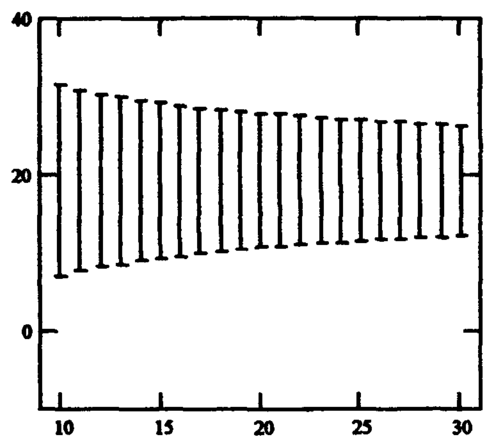


Appendix F.4 cont'd

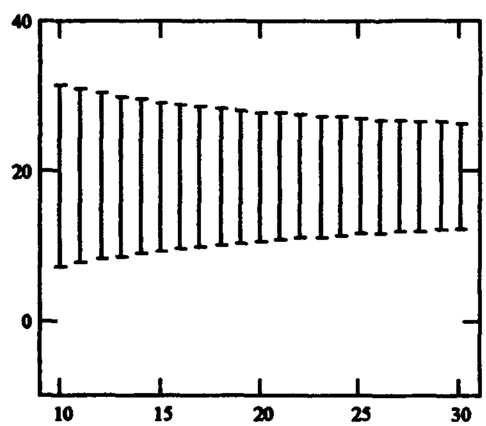
Runtime Group: 10 million minutes



Runtime Group: 20 million minutes



Runtime Group: 30 million minutes

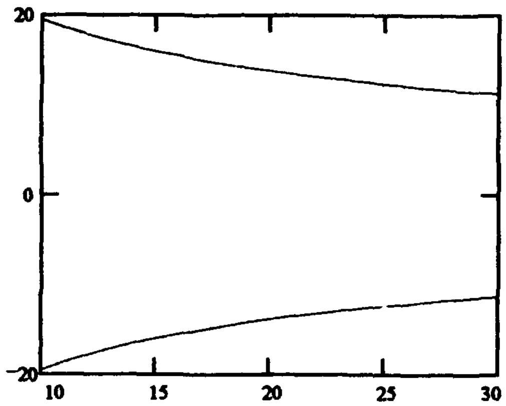


Appendix F.5

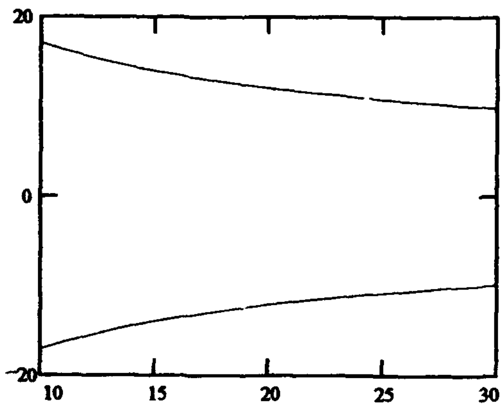
Standard Error of the Differences--
using 10-run averages to calculate the standard error of the difference

Y-axes reflect mean downtimes
X-axes reflect number of runs averaged

Runtime Group: 2 million minutes

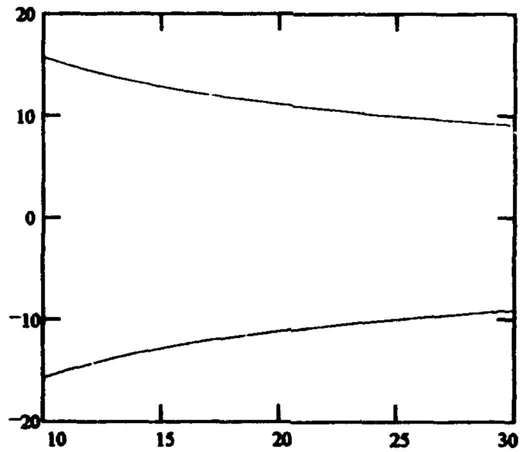


Runtime Group: 5 million minutes

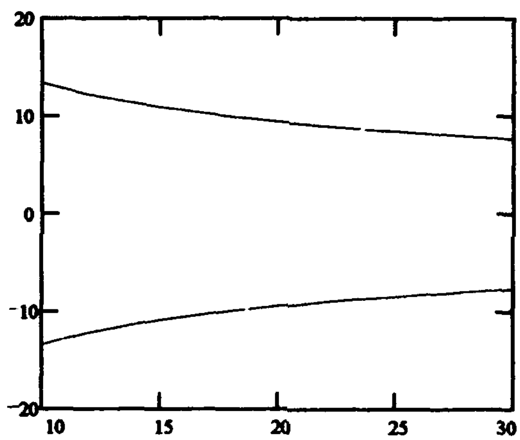


Appendix F.5 cont'd

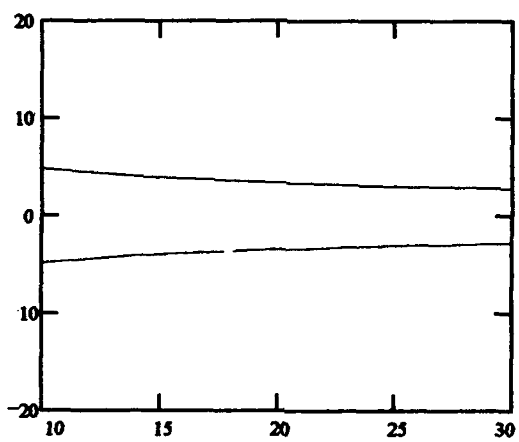
Runtime Group: 10 million minutes



Runtime Group: 20 million minutes



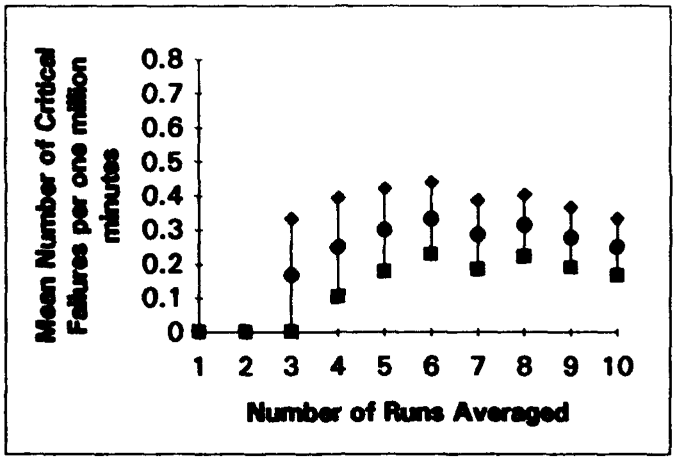
Runtime Group: 30 million minutes



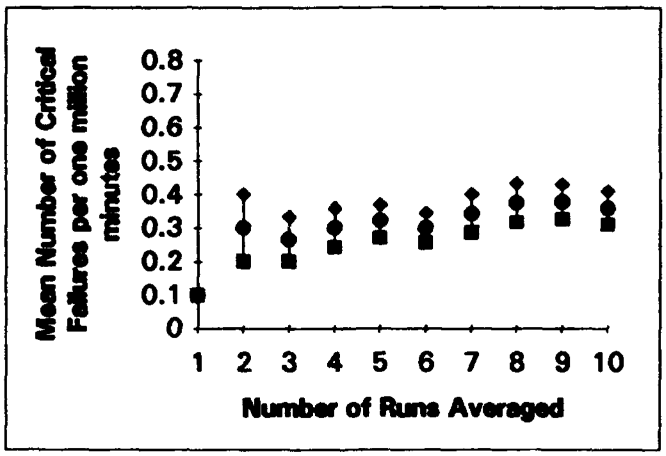
Appendix F.6

Critical Failure Analysis--Standard Error of the Means Plots

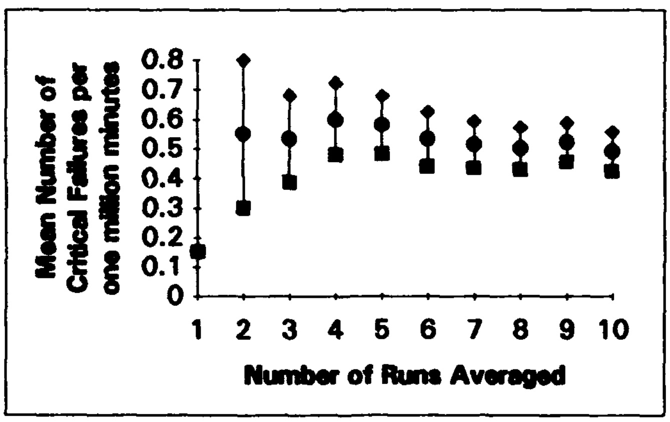
2 Million Minute Runtimes



5 Million Minute Runtimes

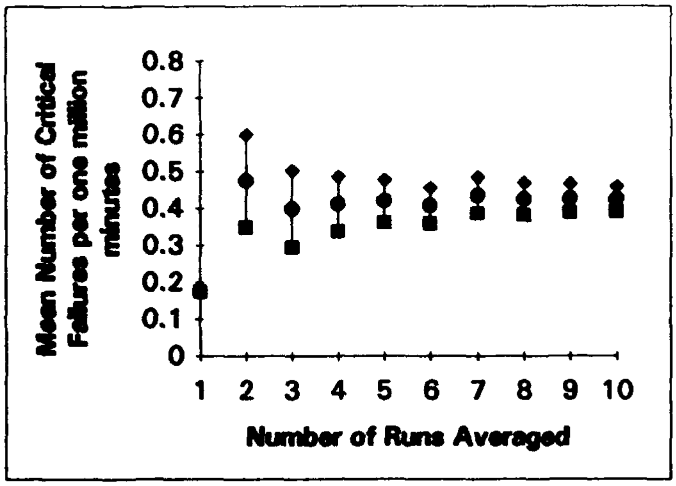


10 Million Minute Runtimes

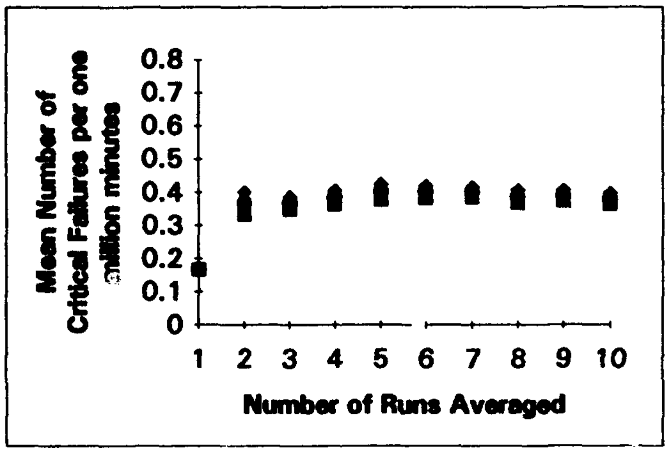


Appendix F.6 cont'd

20 Million Minute Runtimes



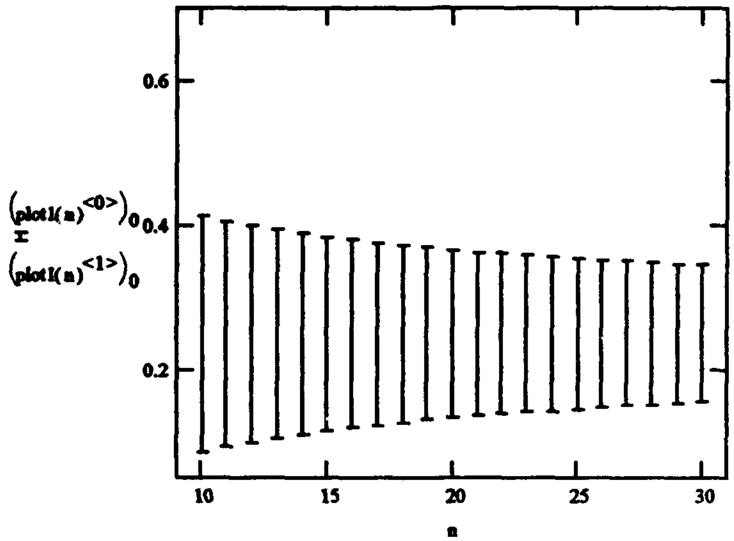
30 Million Minute Runtimes



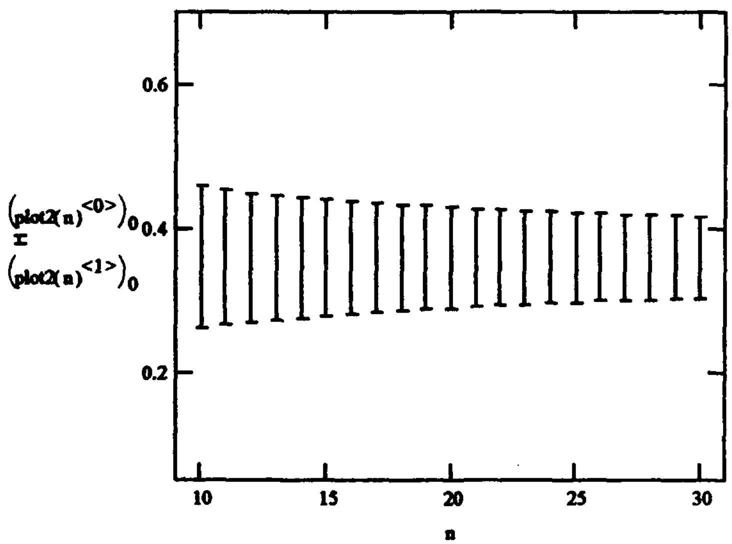
95% Confidence Intervals--
using 10-run averages to calculate the standard error of the mean

Y-axes reflect mean number of critical failures
X-axes reflect number of runs averaged

Runtime Group: 2 million minutes

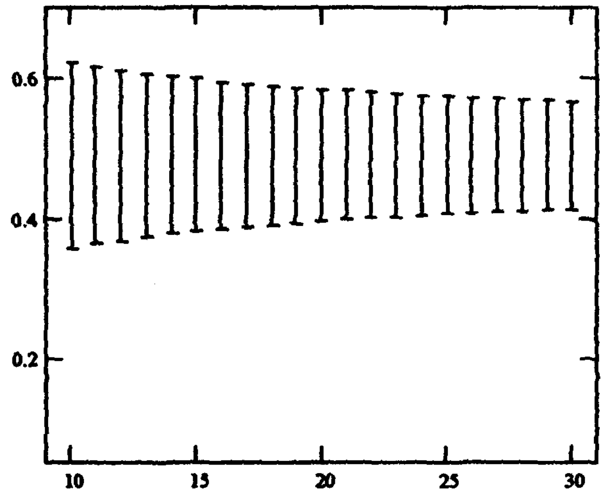


Runtime Group: 5 million minutes

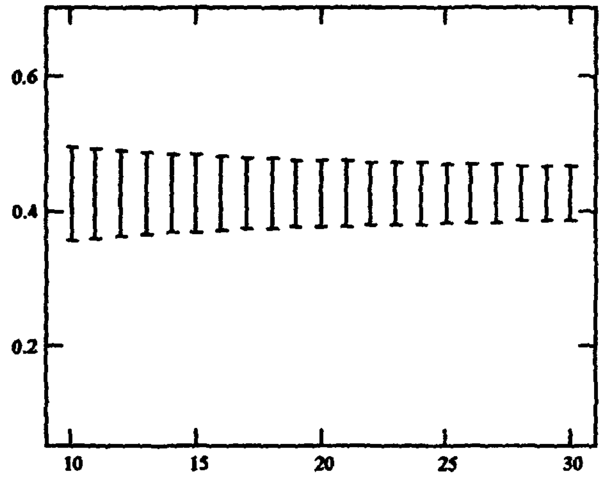


Appendix F.7 cont'd

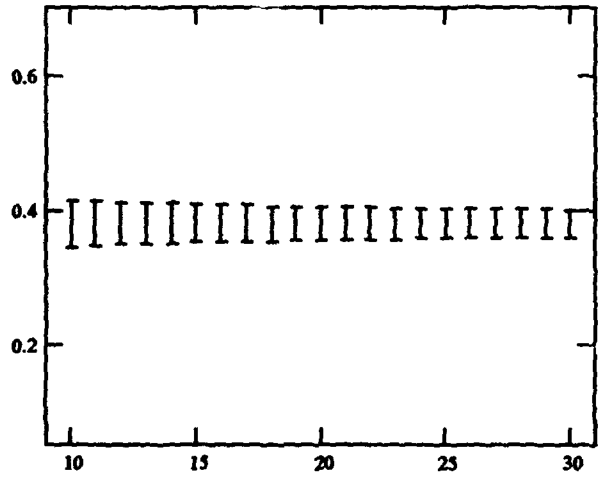
Runtime Group: 10 million minutes



Runtime Group: 20 million minutes



Runtime Group: 30 million minutes

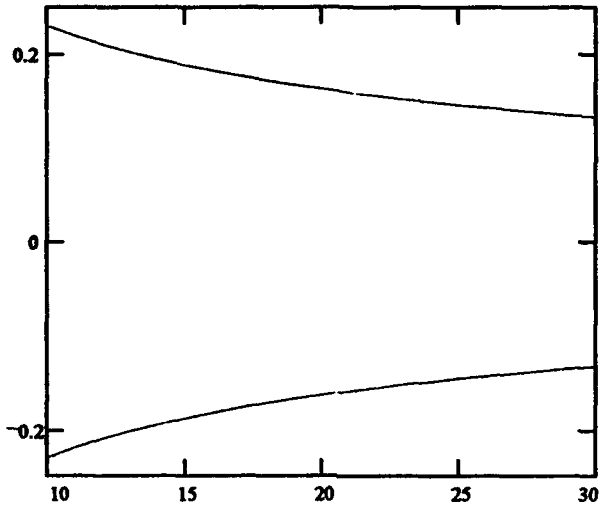


Appendix F.8

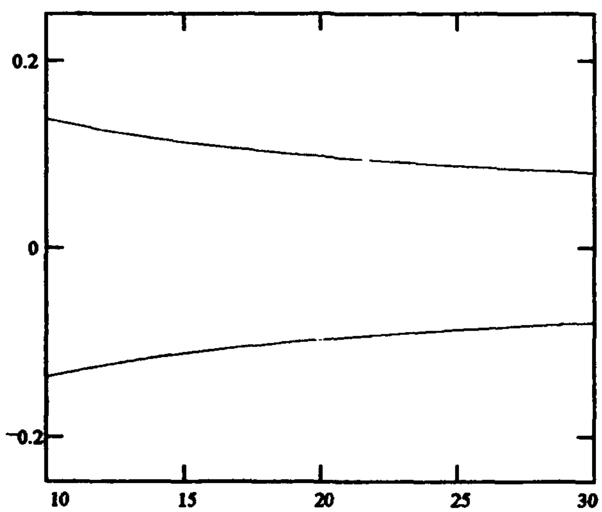
Standard Error of the Differences--
using 10-run averages as a baseline

Y-axes reflect mean number of critical failures
X-axes reflect number of runs averaged

Runtime Group: 2 million minutes

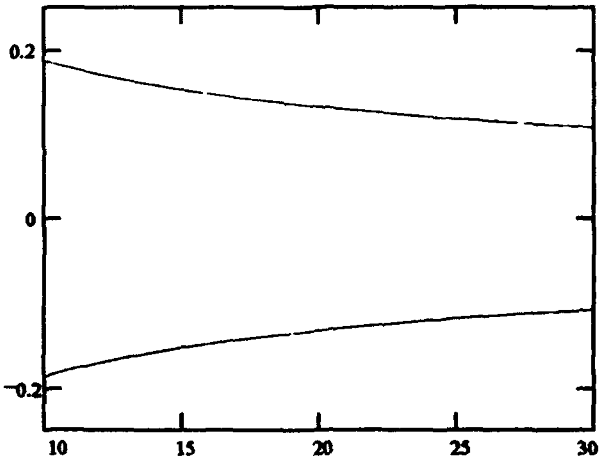


Runtime Group: 5 million minutes

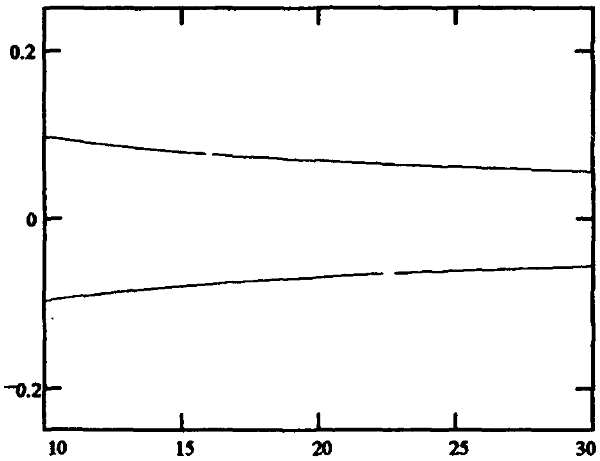


Appendix F.8 cont'd

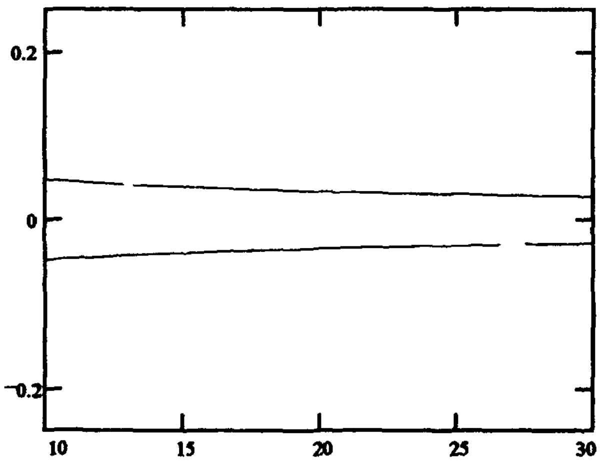
Runtime Group: 10 million minutes



Runtime Group: 20 million minutes



Runtime Group: 30 million minutes



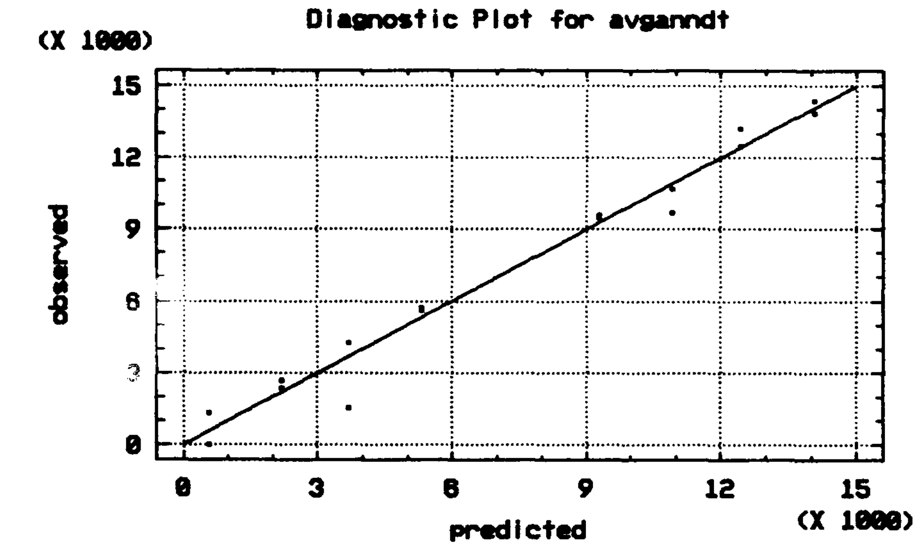
Appendix G.1

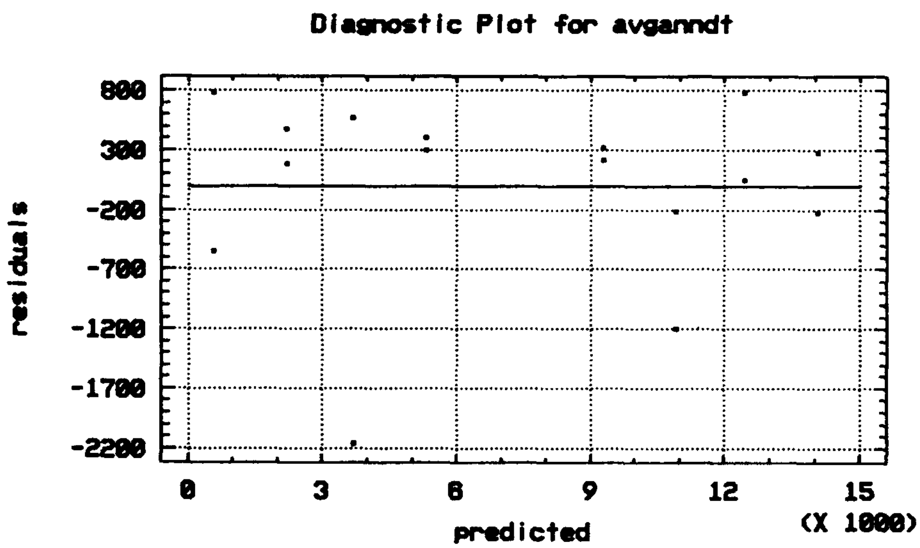
EXPERIMENTAL DESIGN POINTS USED IN RSM ANALYIS OF THE EFFECT OF ELIMINATION OF PART REDUNDANCY ON SYSTEM DOWNTIME

run	GW (#COMP)	AGW (#COMP)	MGW (#COMP)	MP (#COMP)	AM (#COMP)	CP (#COMP)	SM (#COMP)	PDU (#COMP)	HSC (#COMP)
1	-1	-1	-1	-1	1	1	1	1	1
2	1	-1	-1	-1	-1	-1	-1	1	1
3	-1	1	-1	-1	-1	1	1	-1	-1
4	1	1	-1	-1	1	-1	-1	-1	-1
5	-1	-1	1	-1	1	-1	1	-1	1
6	1	-1	1	-1	-1	1	-1	-1	1
7	-1	1	1	-1	-1	-1	1	1	-1
8	1	1	1	-1	1	1	-1	1	-1
9	-1	-1	-1	1	1	1	-1	1	-1
10	1	-1	-1	1	-1	-1	1	1	-1
11	-1	1	-1	1	-1	1	-1	-1	1
12	1	1	-1	1	1	-1	1	-1	1
13	-1	-1	1	1	1	-1	-1	-1	-1
14	1	-1	1	1	-1	1	1	-1	-1
15	-1	1	1	1	-1	-1	-1	1	1
16	1	1	1	1	1	1	1	1	1

run	SC (#COMP)	RA82 (#COMP)	ADOC (#COMP)	NCC (#COMP)	Average Annual Downtime (minutes)
1	1	-1	-1	-1	14336.2656
2	1	1	1	1	2659.5360
3	1	1	1	-1	10704.8960
4	1	-1	-1	1	5628.1248
5	-1	1	-1	1	1536.8544
6	-1	-1	1	-1	9600.0840
7	-1	-1	1	1	1339.2288
8	-1	1	-1	-1	12466.7064
9	-1	-1	1	1	2368.3536
10	-1	1	-1	-1	13832.7408
11	-1	1	-1	1	5724.8352
12	-1	-1	1	-1	9712.0368
13	1	1	1	-1	9499.1688
14	1	-1	-1	1	4265.7696
15	1	-1	-1	-1	13206.7512
16	1	1	1	1	3.6792

Appendix G.2


Diagnostic Plot for avganndt
(X 1000)
15
12
9
6
3
0
observed
0
3
6
9
12
15
predicted
(X 1000)


Diagnostic Plot for avganndt
800
300
-200
-700
-1200
-1700
-2200
residuals
0
3
6
9
12
15
predicted
(X 1000)

Bibliography

1. Bagenstos, Eugene. Granite Sentry Analyst, Martin Marietta Corporation. Personal Interview. 24-25 September, 1993.

2. Bagenstos, Eugene. Granite Sentry Analyst, Martin Marietta Corporation. Telephone Interviews. August 1993 through January 1994.

3. Box, George E. and Norman Draper. *Empirical Model-Building and Response Surfaces*. New York: John Wiley & Sons, Inc., 1987.

4. Boyd, Mark A. and Salvatore J. Bavuso. "Simulation Modeling for Long Duration Spacecraft Control Systems," *Proceedings of the Annual Reliability and Maintainability Symposium*: 106-113, 1993.

5. Brown, Steven O. *An Analysis of the Endurance of Mobile Satellite Command and Control Systems*. MS thesis, AFIT/GOR/ENS/92M-03. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1992 (AAI-9406).

6. D'Agostino, Ralph B. and Michael A. Stephens. *Goodness of Fit Techniques*. New York: Marcel Dekker, Inc., 1986.

7. ESC/USPACECOM. *Final Operational Capability System Specification for the Granite Sentry System (GS-FOC-SS)*. Colorado Springs, CO. 28 September 1992.

8. Goldsman, David. "Simulation Output Analysis", *Proceedings of the 1992 Winter Simulation Conference*: 97-103, 1992.

9. Hachman, Andrew. Scientific Analyst, HQ AFOTEC/SAL, Kirtland AFB, NM. Telephone Interviews. July 1993 through February 1994.

10. -----. Scientific Analyst, HQ AFOTEC/SAL, Kirtland AFB, NM. Electronic Message. 1841hours, 22 July 1993.

11. Martin Marietta Corporation. *Reliability Block Diagram and Math Models Report for the Communications System AN/FYQ-123*. CDRL Sequence No. A020. Colorado Springs, CO. 24 May 1993.

12. -----. *Reliability Prediction and Documentation of Supporting Data for the Communications System AN/FYQ-123*. CDRL Sequence No. A019. Colorado Springs, CO. 26 May 1993.

13. Martin Marietta Corporation/Granite Sentry Integration Team. *Granite Sentry Technical Support Manual: Revision C.* Colorado Springs, CO. 20 May 1992.

14. McGrath, Elgie J. *Techniques for Monte Carlo Simulation. Volume I: Selecting Probability Distributions.* Project No. NR 364-074/1-5-72. Springfield, VA: National Technical Information Service, March 1973 (AD-762721).

15. Miller, Rupert G. *Survival Analysis.* New York: John Wiley & Sons, 1981.

16. Nelson, Barry. "Designing Efficient Simulation Experiments", *Proceedings of the 1992 Winter Simulation Conference*: 127-132, 1992.

17. Palisade Corporation. BestFit manual. Release 1.0. Newfield, NJ. June 1993.

18. Peter, Russel N., Christos Scondras, and Randy Serba. "A Command and Control Application of Evolutionary Acquisition." Report to USAF personnel involved in evolutionary acquisition. February 1991.

19. Pohl, Letitia M. "Evaluation of F-15E Availability During Operational Test", *Proceedings of the 1991 Winter Simulation Conference*: 549-554, 1991.

20. Ross, Sheldon M. *Introduction to Probability Models.* New York: Academic Press, Inc., 1989.

21. Schroeder, G.J. and M. M. Johnson. "Simulation: The correct Approach to Complex Availability Problem", *Proceedings of the 1988 Winter Simulation Conference*: 744-752, 1988.

22. Suchan, Diane et al. AFOTEC/SAL meeting . Personal Interview. 10 June 1993.

23. Whitt, Ward. "The Efficiency of One Long Run Versus Independent Replications in Steady-State Simulation", *Management Science*: 645-665, June 1991.

24. Zacks, Shelemyahu. *Introduction to Reliability Analysis Probability Models and Statistical Methods.* New York: Springer-Verlag, 1992.

Vita

Captain Marilyn J. Bauer was born on 23 January 1967 in Camp Springs, Maryland. She graduated from Loudoun Valley High School, Purcellville, VA in 1985. In September of that year she began studies in the College of Arts and Sciences of the University of Virgina. While attending the University, Capt Bauer completed requirements of the 4-year ROTC program. She graduated in May 1989, receiving a Bachelor of Arts degree in Mathematics and a commission in the United States Air Force. In July 1989, Capt Bauer began her first assignment as a Supply officer in the 347th Fighter Wing Supply Squadron at Moody AFB, GA. Captain Bauer entered AFIT in August of 1992.

Captain Bauer is a member of the Omega Rho International Honor Society.

Permanent address: 11194 Soldier's Ct
Manassas, VA 22110

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1994

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
A Simulation Approach to Granite Sentry System Analysis

5. FUNDING NUMBERS

6. AUTHOR(S)
Marilyn J. Bauer, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/94M-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Capt Hachman
AFOTEC/SAL
Kirtland, AFB NM

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Approved for public release; distribution unlimited

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This study demonstrated the use of simulation modelling to analyze Granite Sentry system performance. The availability simulation model constructed provides a number of system performance measures as a function of component MTBFs and MTTRs. Analysis of failure data prior to model construction supported the generally accepted use of exponentially distributed failure rates and lognormally distributed repair times. A Microsoft Windows version of SLAMSYSTEM proved to be an efficient modelling tool, especially during early stages of model development. Guidelines for model use in system analysis are explored through a runtime analysis and a response surface model of system downtime as a function of part redundancy. The runtime analysis provides recommendations for appropriate simulation runtime and number of replications to produce reasonably efficient and accurate results. The response surface analysis highlights three system components whose part redundancy significantly affects system downtime. Finally, the analytical availability model developed was an essential validation tool in simulation model development.

14. SUBJECT TERMS
Simulation; Availability

15. NUMBER OF PAGES
130

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18
298-102